Computational Analysis of Audio Recordings and Music Scores for the Description and Discovery of Ottoman-Turkish Makam Music

Sertan Şentürk

TESI DOCTORAL UPF / 2016

Director de la tesi

Dr. Xavier Serra Casals
Music Technology Group
Department of Information and Communication Technologies

Universitat
Pompeu Fabra
Barcelona

http://compmusic.upf.edu/senturk2016thesis
http://www.sertansenturk.com/phd-thesis

Dissertation submitted to the Department of Information and Communication Technologies of Universitat Pompeu Fabra in partial fulfillment of the requirements for the degree of

DOCTOR PER LA UNIVERSITAT POMPEU FABRA,

with the mention of *European Doctor*.

Music Technology Group (http://mtg.upf.edu/), Department of Information and Communication Technologies (http://www.upf.edu/etic), Universitat Pompeu Fabra (http://www.upf.edu), Barcelona, Spain

The doctoral defense was held on .... 2017 at Universitat Pompeu Fabra and scored as ....

Dr. Xavier Serra Casals
Thesis Supervisor
Universitat Pompeu Fabra (UPF), Barcelona, Spain

Dr. Gerhard Widmer
Thesis Committee Member
Johannes Kepler University, Linz, Austria

Dr. Barış Bozkurt
Thesis Committee Member
Koç University, Istanbul, Turkey

Dr. Tillman Weyde
Thesis Committee Member
City, University of London, London, United Kingdom

To *Sare*, *Fadime*, *Osman* and *Fatma*, who still show me their boundless love and compassion beyond time...

This thesis has been carried out at the Music Technology Group (MTG) of Universitat Pompeu Fabra (UPF) in Barcelona (Spain) from September 2011 to November 2016, and at the Department of Electrical and Electronics Engineering of Bahçeşehir University in Istanbul (Turkey) from July 2014 to September 2014. It is supervised by Dr. Xavier Serra Casals.

The work in Chapter 6 has been partially conducted in close collaboration with Dr. André Holzapfel (KTH Royal Institute of Technology, Stockholm, Sweden). Dr. M. Kemal Karaosmanoğlu (Yıldız University, Istanbul, Turkey) has been consulted in all matters pertaining the music scores used throughout the work.

The work in several parts of this thesis have been carried out in collaboration with the CompMusic team at the MTG, the partner group led by Dr. Barış Bozkurt (Koç University, Istanbul, Turkey) and the Perceptual Intelligent Lab – Statistical Inference Group led by Dr. Ali Taylan Cemgil (Boğaziçi University, Istanbul, Turkey). A detailed list of collaborators include (alphabetically ordered) Alastair Porter, Altuğ Karakurt, Andrés Ferraro, Burak Uyar, Hasan Sercan Atlı, Gopala Krishna Koduri, Miraç Atıcı, Mohamed Sordo, Sankalp Gulati and Uğur Şimşekli. M. Kemal Karaosmanoğlu and Nezihe Şentürk (Gazi University, Ankara, Turkey) have been consulted for all the music related aspects in the work.

This work has been supported by the Department of Information and Communication Technologies (DTIC) PhD fellowship (2011–16), Universitat Pompeu Fabra and the European Research Council (ERC) under the European Union's Seventh Framework Program, as part of the CompMusic project (ERC grant agreement 267583). The research stay at the Bahçeşehir University, Istanbul has been funded by European Commission under the Erasmus+ program.

The epigraphs in the beginning of each Chapter are taken from İhsan Oktay Anar's phenomenal novel *Suskunlar (Taciturns)*.

Acknowledgements

Phew, what a journey it was..!

It has been a little bit more that 5 years since I have started my journey in the CompMusic project. Seasons have passed, lives have changed. The world has *really* changed (I wonder how Dickens would have started its tale). I have also changed, for better or worse. This thesis is a fruit of these years; a fruit which I think as the harvest of collaborative labor.

First of all, I would like to thank my supervisor Xavier Serra. It is his vision, which made the CompMusic project possible. It is his perspective, which gathered this exceptional team throughout the world. Likewise, it is his guidance that shaped the thesis. There is not a single day I find myself thinking of all merits I have learned from him and how much more there is still to gain. He has the biggest role in what I have become as a determined researcher, for which I owe the greatest gratitute.

Nevertheless, I should say that I owe the biggest debt to M. Kemal Karaosmoğlu. Without his unparalleled devotion to collect the makam music scores for decades, and his invaluable help throughout the thesis, this work would simply be impossible. He has been and will always be the definition of idealism I would like to take an example of.

I was able to realize most of the audio-score alignment, the core contribution of my thesis, under André Holzapfel's guidance.

Moreover, he has given me some of the most crucial and sincere suggestions about my PhD work. He am indebted to him not only for his lead during my PhD, and but for his sincerity as a friend.

Andres Ferrero is definitely one of the most fruitful people I have had the chance to work with. It is his tireless effort and dedication, which made our "baby" web application become what it is.

I would like to give a special thanks to Ara Dinkjian, Özer Özel, Robert Garfias and Harold Hagopian for granting us the streaming rights of many recordings they hold the rights of. Seymour Shilen has also helped me numerous times by pointing out the errors in the content.

Last but not the least, I want to thank all my family (especially my mother, father and my *kel* sister) and my friends (especially Emre Can Dağlıoğlu and Can Öktemer). Nothing has changed in the last five years: "Without their selfless support, I would have never even made it to a new day. On the contrary, without my ridiculous ways, they would have a much better life." Müge Kalender has always been on my side for all these uncountable days and nights, until the very last moment. I don't know how many times Levent Ateşoğlu took care of me. I will always be grateful to Eva Arrizabalaga: I would not be able to continue without her motherly care, while all the bombs (literally) were exploding. Muchos besos a Tamar Nalcı, Narod Erkol, Beril Karavit, Cana Göksu, Berta Garcia Faet, Dafni Anastasiadi and Rüstem Can. Finally, I should not forget Şeker and İrfan Hamdi, for their ... cuteness.

Abstract

This thesis addresses several shortcomings on the current state of the art methodologies in music information retrieval (MIR). In particular, it proposes several computational approaches to automatically analyze and describe music scores and audio recordings of Ottoman-Turkish makam music (OTMM). The main contributions of the thesis are the music corpus that has been created to carry out the research and the audio-score alignment methodology developed for the analysis of the corpus. In addition, several novel computational analysis methodologies are presented in the context of common MIR tasks of relevance for OTMM. Some example tasks are predominant melody extraction, tonic identification, tempo estimation, makam recognition, tuning analysis, structural analysis and melodic progression analysis. These methodologies become a part of a complete system called *Dunya-makam* for the exploration of large corpora of OTMM.

The thesis starts by presenting the created *CompMusic Ottoman-Turkish makam music corpus*. The corpus includes 2200 music scores, more than 6500 audio recordings, and accompanying metadata. The data has been collected, annotated and curated with the help of music experts. Using criteria such as *completeness*, *coverage* and *quality*, we validate the corpus and show its research potential. In fact, our corpus is the largest and most representative resource of OTMM that can be used for computational research. Several test datasets have also been created from the corpus to develop and evaluate the specific methodologies proposed for different computational tasks addressed in the thesis.

The part focusing on the analysis of music scores is centered on phrase and section level structural analysis. Phrase boundaries are automatically identified using an existing state-of-the-art segmentation methodology.

Section boundaries are extracted using heuristics specific to the formatting of the music scores. Subsequently, a novel method based on graph analysis is used to establish similarities across these structural elements in terms of melody and lyrics, and to label the relations semiotically.

The audio analysis section of the thesis reviews the state-of-the-art for analysing the melodic aspects of performances of OTMM. It proposes adaptations of existing predominant melody extraction methods tailored to OTMM. It also presents improvements over pitch-distribution-based tonic identification and makam recognition methodologies.

The audio-score alignment methodology is the core of the thesis. It addresses the culture-specific challenges posed by the musical characteristics, music theory related representations and oral praxis of OTMM. Based on several techniques such as subsequence dynamic time warping, Hough transform and variable-length Markov models, the audio-score alignment methodology is designed to handle the structural differences between music scores and audio recordings. The method is robust to the presence of non-notated melodic expressions, tempo deviations within the music performances, and differences in tonic and tuning. The methodology utilizes the outputs of the score and audio analysis, and *links* the audio and the symbolic data. In addition, the alignment methodology is used to obtain score-informed description of audio recordings. The score-informed audio analysis not only simplifies the audio feature extraction steps that would require sophisticated audio processing approaches, but also substantially improves the performance compared with results obtained from the state-of-the-art methods solely relying on audio data.

The analysis methodologies presented in the thesis are applied to the *CompMusic Ottoman-Turkish makam music corpus* and integrated into a web application aimed at culture-aware music discovery. Some of the methodologies have already been applied to other music traditions such as Hindustani, Carnatic and Greek music. Following open research best practices, all the created data, software tools and analysis results are openly available. The methodologies, the tools and the corpus itself provide vast opportunities for future research in many fields such as music information retrieval, computational musicology and music education.

Özet

Osmanlı-Türk Musikisinin Betimlenmesi ve Keşfi için Ses Kayıtlarının ve Basılı Notaların Hesaplamalı Analizi

Bu tez, müzik bilgi erişim alanının günümüzdeki en gelişkin yöntemlerinin çeşitli eksikliklerine odaklanmaktadır. Tez kapsamında özellikle Osmanlı-Türk makam musikisi (OTMM) notalarının ve ses kayıtlarının otomatik analiz edilebilmesi ve betimlenebilmesi için çeşitli hesaplamalı yaklaşımlar önerilmiştir. Tezin başlıca katkıları arasında araştırmaların yapılabilmesi için oluşturulan derlem ve derlemin analiz edilebilmesi için geliştirilen icra-nota eşleme yöntemi yer almaktadır. Bunlara ek olarak, OTMM ile ilişkili müzik bilgi erişim konuları için yeni hesaplamalı yöntemler de önerilmiştir. Baskın ezgi analizi, karar perdesi tespiti, tempo kestirimi, makam tanıma, perde analizi, yapısal analiz ve seyir analizi bu konulara örnekler olarak verilebilir. Bu yöntemler, OTMM için hazırlanan derlemleri incelemek üzere geliştirilen *Dunya-makam* adındaki sistemin bir parçası olmuştur.

Tez, oluşturulan *OTMM derlemi*ni tanıtarak başlamaktadır. Derlem, bünyesinde 2200 basılı nota, 6500' den fazla ses kaydı ve bunlarla ilgili meta-verileri barındırmaktadır. Tüm veriler sözü geçen müziğin uzmanlarının yardımlarıyla toparlanmış, işaretlenmiş ve düzenlenmiştir. Oluşturulan derlemin olası araştırmalar için uygunluğu bütünlük, kapsam ve kalite gibi ölçütler göz önünde bulundurularak doğrulanmıştır. Esasında *OTMM derlemi* hesaplamalı araştırmalar için hazırlanmış halihazırdaki en kapsamlı ve kültürü en iyi biçimde yansıtan derlemdir. Ayrıca tez kapsamında belirtilen hesaplamalı araştırma konuları için hazırlanan der-

lemden yararlanarak birçok farklı deney veri kümesi oluşturulmuştur.

Basılı notalarının analizi ile ilgili kısım, cümle ve bölüm düzeyinde yapısal analiz üzerinde yoğunlaşmıştır. Daha önce geliştirilmiş olan bir bölütleme yöntemi ile cümlelerin sınırları otomatik olarak tespit edilmektedir. Bölüm sınırları, basılı notalardaki kısmi işaretlemelerden otomatik olarak çıkarılmaktadır. Ardından, grafiksel analize dayalı özgün bir yöntem, bu yapıların ezgisel ve güftesel benzerliklerini kurmak ve bu ilişkileri göstergesel olarak etiketlemek için kullanılmaktadır.

Tezin ses kaydı analizi kısmı OTMM icralarının ezgisel yapılarına yönelik var olan güncel yöntemlerin değerlendirilmesini içermektedir ve mevcut baskın ezgi analizi yöntemlerinin OTMM için nasıl uygun hale getirildiğini anlatmaktadır. Ayrıca bu bölümde, perde dağılımına dayalı otomatik karar perdesi tespiti ve makam tanıma yöntemlerindeki gelişmeler de açıklanmaktadır.

Nota-icra eşleştirme yöntemi tezin temelini oluşturmaktadır. Bu yöntem, OTMM' nin kültüre özgü olan müzikal karakteristiğine, müzik kuramıyla ilişkili gösterimlerine ve sözel alışkanlıklarına dayalı zorluklara uygun tasarlanmıştır. Dinamik zaman yamultması, Hough dönüşümü ve değişken-uzunluklu Markov modelleri gibi birçok tekniğe dayandırılarak geliştirilen yeni yöntem, farklı yapısal özelliklerde bulunan icralarla notaları eşleştirebilmektedir. Yöntem, basılı notalarda belirtilmeyen ezgisel anlatımlardan, icra içerisindeki tempo değişimlerinden ve farklı ahenklerden etkilenmemektedir. Ayrıca, bu yöntem ses kaydı analiz sonuçlarından ve notalardan faydalanarak sembolik veriler ile ses kaydı arasındaki bağlantıları kurmaktadır. Bunlara ek olarak, eşleştirme yöntemi, ses kayıtlarının notadan yararlanarak açıklanmasında da kullanılmaktadır. Notadan yararlanarak elde edilen ses kaydı özellikleri, özelliklerin çıkarılması aşamasında karmaşık ses işleme yaklaşımlarına ihtiyaç duyabilecek adımları daha kolay hale getirmektedir. Aynı zamanda yürürlükteki yalnızca ses verisine dayanan yöntemlerin verimini de gözle görülür bir biçimde artırmıştır.

Bu tezde sunulan yöntemlerin tümü, *CompMusic—OTMM derlemi*ne uygulanmış ve web tabanlı kültürel farkındalığı olan (*culture-aware*) bir müzik keşif uygulamasıyla entegre edimiştir. Tez kapsamında sunulan bazı yöntemler, Hindustani, Karnatik ve Yunan müziklerine de uygulanmıştır. Açık-araştırma kavramının en iyi uygulamalarını izleyerek tez kapsamında derlenen tüm veriler, yazılım araçları ve analiz sonuçları paylaşıma açılmıştır. Sunulan yöntemler, araçlar ve derlem, müzik bilgi erişim, hesaplamalı müzikoloji ve müzik eğitimi gibi alanlarda yapılacak araştırmalara yararlı olabilecektir.

İngilizce'den Türkçe'ye Hasan Sercan Atlı tarafından çevrilmiştir.

Resum

Anàlisis Computacional d'Enregistraments d'Àudio i Partitures Musicals per a la Descripció i Exploració de Música Makam Turca Otomana

Aquesta tesi adreça diverses deficiències en l'estat actual de les metodologies d'extracció d'informació de música (Music Information Retrieval o MIR). En particular, la tesi proposa diverses estratègies per analitzar i descriure automàticament partitures musicals i enregistraments d'actuacions musicals de música Makam Turca Otomana (OTMM en les seves sigles en anglès). Les contribucions principals de la tesi són els corpus musicals que s'han creat en el context de la tesi per tal de dur a terme la recerca i la metodologia de alineament d'àudio amb la partitura que s'ha desenvolupat per tal d'analitzar els corpus. A més la tesi presenta diverses noves metodologies d'anàlisi computacional d'OTMM per a les tasques més habituals en MIR. Alguns exemples d'aquestes tasques són la extracció de la melodia principal, la identificació del to musical, l'estimació de tempo, el reconeixement de Makam, l'anàlisi de la afinació, l'anàlisi de la estructura musical i l'anàlisi de la progressió melòdica. Aquest seguit de metodologies formen part del sistema *Dunya-makam* per a la exploració de grans corpus musicals d'OTMM.

En primer lloc, la tesi presenta el *corpus CompMusic Ottoman-Turkish makam music*. Aquest inclou 2200 partitures musicals, més de 6500 enregistraments d'àudio i metadata complementària. Les dades han sigut recopilades i anotades amb ajuda d'experts en aquest repertori mu-

sical. El corpus ha estat validat en termes de *d'exhaustivitat*, *cobertura* i *qualitat* i mostrem aquí el seu potencial per a la recerca. De fet, aquest corpus és el la font més gran i representativa de OTMM que pot ser utilitzada per recerca computacional. També s'han desenvolupat diversos subconjunts de dades per al desenvolupament i evaluació de les metodologies específiques proposades per a les diverses tasques computacionals que es presenten en aquest tesi.

La secció de la tesi que tracta de l'anàlisi de partitures musicals se centra en l'anàlisi estructural a nivell de secció i de frase musical. Els límits temporals de les frases musicals s'identifiquen automàticament gràcies a un metodologia de segmentació d'última generació. Els límits de les seccions s'extreuen utilitzant un seguit de regles heurístiques determinades pel format de les partitures musicals. Posteriorment s'utilitza un nou mètode basat en anàlisi gràfic per establir semblances entre aquest elements estructurals en termes de melodia i text. També s'utilitza aquest mètode per etiquetar les relacions semiòtiques existents.

La següent secció de la tesi tracta sobre anàlisi d'àudio i en particular revisa les tecnologies d'avantguardia d'anàlisi dels aspectes melòdics en OTMM. S'hi proposen adaptacions dels mètodes d'extracció de melodia existents que s'ajusten a OTMM. També s'hi presenten millores en metodologies de reconeixement de makam i en identificació de tònica basats en distribució de to.

La metodologia d'alineament d'àudio amb partitura és el nucli de la tesi. Aquesta aborda els reptes culturalment específics imposats per les característiques musicals, les representacions de la teoria musical i la pràctica oral particulars de l'OTMM. Utilitzant diverses tècniques tal i com Dynamic Time Warping, Hough Transform o models de Markov de durada variable, la metodologia d'alineament esta dissenyada per enfrontar les diferències estructurals entre partitures musicals i enregistraments d'àudio. El mètode és robust inclús en presència d'expressions musicals no anotades en la partitura, desviacions de tempo ocorregudes en les actuacions musicals i diferències de tònica i afinació. La metodologia aprofita els resultats de l'anàlisi de la partitura i l'àudio per *enllaçar* la informació simbòlica amb l'àudio. A més, la tècnica d'alineament s'utilitza per obtenir descripcions de l'àudio fonamentades en la partitura. L'anàlisi de l'àudio fonamentat en la partitura no només simplifica les fases d'extracció de característiques d'àudio que requeririen de mètodes de processament d'àudio sofisticats, sinó que a més millora substancialment els resultats comparat amb altres mètodes d´ultima generació que només depenen de contingut d'àudio.

Les metodologies d'anàlisi presentades s'han utilitzat per analitzar el *corpus CompMusic Ottoman-Turkish makam music* i s'han integrat en una aplicació web destinada al descobriment musical de tradicions culturals específiques. Algunes de les metodologies ja han sigut també aplicades a altres tradicions musicals com la Hindustani, la Carnàtica i la Grega. Seguint els preceptes de la investigació oberta totes les dades creades, eines computacionals i resultats dels anàlisis estan disponibles obertament. Tant les metodologies, les eines i el corpus en si mateix proporcionen àmplies oportunitats per recerques futures en diversos camps de recerca tal i com la musicologia computacional, la extracció d'informació musical i la educació musical.

Traducció d'anglès a català per Oriol Romaní Picas.

Resumen

Análisis Computacional de Grabaciones de Audio y Partituras para la Descripción y Descubrimiento de Música de Makam Turco-Otomana

Esta tesis aborda varias limitaciones de las metodologías más avanzadas en el campo de recuperación de información musical (MIR por sus siglas en inglés). En particular, propone varios métodos computacionales para el análisis y la descripción automáticas de partituras y grabaciones de audio de música de makam turco-otomana (MMTO). Las principales contribuciones de la tesis son el corpus de música que ha sido creado para el desarrollo de la investigación y la metodología para alineamiento de audio y partitura desarrollada para el análisis del corpus. Además, se presentan varias metodologías nuevas para análisis computacional en el contexto de las tareas comunes de MIR que son relevantes para MMTO. Algunas de estas tareas son, por ejemplo, extracción de la melodía predominante, identificación de la tónica, estimación de tempo, reconocimiento de makam, análisis de afinación, análisis estructural y análisis de progresión melódica. Estas metodologías constituyen las partes de un sistema completo para la exploración de grandes corpus de MMTO llamado *Dunya-makam*.

La tesis comienza presentando el *corpus de música de makam turco-otomana de CompMusic*. El corpus incluye 2200 partituras, más de 6500 grabaciones de audio, y los metadatos correspondientes. Los datos han sido recopilados, anotados y revisados con la ayuda de expertos. Utilizan-

do criterios como *compleción*, *cobertura* y *calidad*, validamos el corpus y mostramos su potencial para investigación. De hecho, nuestro corpus constituye el recurso de mayor tamaño y representatividad disponible para la investigación computacional de MMTO. Varios conjuntos de datos para experimentación han sido igualmente creados a partir del corpus, con el fin de desarrollar y evaluar las metodologías específicas propuestas para las diferentes tareas computacionales abordadas en la tesis.

La parte dedicada al análisis de las partituras se centra en el análisis estructural a nivel de sección y de frase. Los márgenes de frase son identificados automáticamente usando uno de los métodos de segmentación existentes más avanzados. Los márgenes de sección son extraídos usando una heurística específica al formato de las partituras. A continuación, se emplea un método de nueva creación basado en análisis gráfico para establecer similitudes a través de estos elementos estructurales en cuanto a melodía y letra, así como para etiquetar relaciones semióticamente.

La sección de análisis de audio de la tesis repasa el estado de la cuestión en cuanto a análisis de los aspectos melódicos en grabaciones de MMTO. Se proponen modificaciones de métodos existentes para extracción de melodía predominante para ajustarlas a MMTO. También se presentan mejoras de metodologías tanto para identificación de tónica basadas en distribución de alturas, como para reconocimiento de makam.

La metodología para alineación de audio y partitura constituye el grueso de la tesis. Aborda los retos específicos de esta cultura según vienen determinados por las características musicales, las representaciones relacionadas con la teoría musical y la praxis oral de MMTO. Basada en varias técnicas tales como deformaciones dinámicas de tiempo subsecuentes, transformada de Hough y modelos de Markov de longitud variable, la metodología de alineamiento de audio y partitura está diseñada para tratar las diferencias estructurales entre partituras y grabaciones de audio. El método es robusto a la presencia de expresiones melódicas no anotadas, desviaciones de tiempo en las grabaciones, y diferencias de tónica y afinación. La metodología utiliza los resultados del análisis de partitura y audio para *enlazar* el audio y los datos simbólicos. Además, la metodología de alineación se usa para obtener una descripción informada por partitura de las grabaciones de audio. El análisis de audio informado por partitura no sólo simplifica los pasos para la extracción de características de audio que de otro modo requerirían sofisticados métodos de procesado de audio, sino que también mejora sustancialmente su rendimiento en comparación con los resultados obtenidos por los métodos más avanzados basados únicamente en datos de audio.

Las metodologías analíticas presentadas en la tesis son aplicadas al *corpus de música de makam turco-otomana de CompMusic* e integradas en una aplicación web dedicada al descubrimiento culturalmente específico de música. Algunas de las metodologías ya han sido aplicadas a otras tradiciones musicales, como música indostaní, carnática y griega. Siguiendo las mejores prácticas de investigación en abierto, todos los datos creados, las herramientas de software y los resultados de análisis está disponibles públicamente. Las metodologías, las herramientas y el corpus en sí mismo ofrecen grandes oportunidades para investigaciones futuras en muchos campos tales como recuperación de información musical, musicología computacional y educación musical.

Traducción de inglés a español por Rafael Caro Repetto.

Contents

List of Symbols

The following is a list of different symbols used in the dissertation along with a short description of each symbol.

Symbol	Description
	General symbols
$\{.\}$	Set
$[.]$	Sequence
$< . >$	Tuple
x	An arbitrary “object.” Can be a document (a music score or an audio recording); an element (a section, phrase, measure, note etc.) in a document; a feature extracted from another document; a model; or also a (sub)set or (sub)sequence of objects.
$\|\boldsymbol{x}\|$	The number of elements in the set or the subsequence (x).
$t(x)$	Time interval or timestamp (for instantaneous events) of the object (x).
$t_{ini}(x)$	Initial timestamp of the object (x).
$t_{fin}(x)$	Final timestamp of the object (x).
$d(x)$	Duration of the object (x).
$\bar{f}^{(x)}$	An arbitrary fragment selected from an object, e.g. the performance between 50th and 70th seconds of an audio recording.

Symbol	Description
$f^{(x)}$	The label of an arbitrary fragment selected from an object, e.g. interval between the 50^{th} and 70^{th} seconds of an audio recording.
$\bar{\boldsymbol{F}}^{(x)}$	The set of fragments in an object (x).
$\mathbf{N}^{(x)}$	Note symbol sequence in the object (x).
$\mathbf{S}^{(x)}$	Section symbol sequence in the object (x).
$n_i^{(x)}$	i^{th} note symbol in the note symbol sequence $\mathbf{N}^{(x)}$.
$s_i^{(x)}$	i^{th} section symbol in the note symbol sequence $\mathbf{S}^{(x)}$.
$\bar{\mathbf{N}}^{(x)}$	Note sequence in the object (x).
$\bar{\mathbf{M}}^{(x)}$	Measure sequence in the object (x).
$\bar{\mathbf{S}}^{(x)}$	Section sequence in the object (x).
$\bar{\mathbf{P}}^{(x)}$	Phrase sequence in the object (x).
$\bar{n}_i^{(x)}$	i^{th} note in the object (x).
$\bar{m}_i^{(x)}$	i^{th} measure in the object (x).
$\bar{s}_i^{(x)}$	i^{th} section in the object (x).
$\bar{p}_i^{(x)}$	i^{th} phrase in the object (x).
$\mathcal{S}_s{}^{(x)}$	Set of section symbols in the object (x). Equals to $\left\{\mathbf{S}^{(x)}\right\}$.
$\mathcal{S}^{(x)}$	Set of section symbols in the object (x) plus an $unrelated$ element.
$\boldsymbol{\lambda}^{(x)}$	The lyrics (sequence) associated with the object (x).
$\boldsymbol{\mathcal{G}}$	A graph.
$\boldsymbol{\mathcal{N}}(\boldsymbol{\mathcal{G}})$	Nodes of a graph $\boldsymbol{\mathcal{G}}$.
$\boldsymbol{\mathcal{E}}(\boldsymbol{\mathcal{G}})$	Edges of a graph $\boldsymbol{\mathcal{G}}$.
$e_{i\to j}$	An edge from the i^{th} to the j^{th} node.
$\boldsymbol{\mathcal{P}}(\boldsymbol{\mathcal{G}})$	Set of paths extracted from the graph $\boldsymbol{\mathcal{G}}$.
$\mathbf{p}$	A path formed by a sequence of nodes and connecting edges.

Symbol	Description
	Research Corpus
$\mathscr{A}^{(\boldsymbol{\mathcal{R}})}$	The subjected attribute set (makams, forms or usuls) of the collection $\boldsymbol{\mathcal{R}}$.
$\alpha_k^{(\boldsymbol{\mathcal{R}})}$	An attribute (makam, form or usul) in the collection $\boldsymbol{\mathcal{R}}$.
$\mathscr{O}\left(\mathscr{A}^{(\boldsymbol{\mathcal{R}})}\right)$	The overlap of the set of the subjected attribute set $\mathscr{A}^{(\textbf{SymbTr})}$ of the **SymbTr** collection with respect to the set of the subjected attribute set $\mathscr{A}^{(\boldsymbol{\mathcal{R}})}$ of the reference collection $\boldsymbol{\mathcal{R}}$.
$o\left(\alpha_k^{(\boldsymbol{\mathcal{R}})}\right)$	The occurence count of the attribute $\alpha_k^{(\boldsymbol{\mathcal{R}})} \in \mathscr{A}^{(\boldsymbol{\mathcal{R}})}$.
$\hat{o}\left(\alpha_k^{(\boldsymbol{\mathcal{R}})}\right)$	The occurence count ratio of the attribute $\alpha_k^{(\boldsymbol{\mathcal{R}})} \in \mathscr{A}^{(\boldsymbol{\mathcal{R}})}$.
$\hat{O}\left(\alpha_k^{(\boldsymbol{\mathcal{R}})}\right)$	The cumulative occurrence ratio of the attribute $\alpha_k^{(\boldsymbol{\mathcal{R}})} \in \mathscr{A}^{(\boldsymbol{\mathcal{R}})}$.
$\mathscr{O}_k\left(\mathscr{A}^{(\boldsymbol{\mathcal{R}})}\right)$	The overlap of the **SymbTr** collection against the cumulative occurrence ratio $\hat{O}\left(\alpha_k^{(\boldsymbol{\mathcal{R}})}\right)$ of the attribute $\alpha_k^{(\boldsymbol{\mathcal{R}})} \in \mathscr{A}^{(\boldsymbol{\mathcal{R}})}$ of the reference collection $\boldsymbol{\mathcal{R}}$.
$\mathscr{C}(\mathscr{A}^{(\boldsymbol{\mathcal{R}})})$	Attribute coverage of the collection $\boldsymbol{\mathcal{R}}$ by **SymbTr** collection.
	Score Analysis
$\hat{\boldsymbol{\Psi}}^{(b)}$	Synthetic melody computed from a music score fragment (b).
$\hat{\psi}_i^{(b)}$	i^{th} pitch sample in the synthetic melody $\hat{\boldsymbol{\Psi}}^{(b)}$computed from the score fragment (b).
$\hat{\boldsymbol{\Omega}}^{(x)}$	Synthetic harmonic pitch class profiles (HPCPs) (Gómez, 2006) computed from the score fragment (x) in MIDI format.
$\hat{\omega}_i^{(x)}$	i^{th} bin of the synthetic HPCPs computed from the score fragment (x) in MIDI format.
$\boldsymbol{\mathcal{U}}$	The set of unique cliques.
$\boldsymbol{\mathcal{V}}$	The set of similar cliques.

Symbol	Description
$\mathcal{W}$	The set of all intersections between different similar cliques.
u	A unique clique.
v	A similar clique.
w	An intersection between different similar cliques.
$\mathscr{L}(\mathbf{x}, \mathbf{y})$	Levenshtein distance between two strings **x** and **y**.
$\hat{\mathscr{L}}(\mathbf{x}, \mathbf{y})$	Normalized Levenshtein distance between two strings **x** and **y**.
l	Similarity threshold used in score structural analysis.
$\Lambda_{mel}^{(x)}$	Semiotic melody label of an arbitrary clique or structural element (x) in a music score.
$\Lambda_{lyr}^{(x)}$	Semiotic lyrics label of an arbitrary clique or structural element (x) in a music score.
	Audio Analysis
$\boldsymbol{\varrho}^{(a)}$	Predominant melody extracted from an audio fragment (a).
$\rho_i^{(a)}$	i^{th} pitch sample of the predominant melody $\boldsymbol{\varrho}^{(a)}$extracted from the audio fragment (a).
$\hat{\boldsymbol{\varrho}}^{x,(a)}$	Predominant melody extracted from an audio fragment (a) and normalized with respect to the reference frequency x.
$\hat{\rho}_i^{x,(a)}$	i^{th} pitch sample of the predominant melody $\hat{\boldsymbol{\varrho}}^{x,(a)}$, extracted from the audio fragment (a) and normalized with respect to the reference frequency x.
$\hat{\boldsymbol{\Gamma}}^{x,(a)}$	Harmonic pitch class profile (Gómez, 2006) (HPCP) extracted from an audio fragment (a). The first is centered around the pitch-class of x.
$\hat{\gamma}_i^{x,(a)}$	i^{th} bin of the HPCPs extracted from an audio fragment (a). The first bin is centered around the pitch-class of x.

Symbol	Description
$\triangle(x, y)$	“Octave-wrapped” cent distance from the frequency x to y.
$\blacktriangle(x, y)$	Shortest “octave-wrapped” cent distance between the frequencies x to y.
$\boldsymbol{H}^{(a)}$	Pitch distribution (PD) or pitch-class distribution (PCD), extracted from the audio fragment (a). The bins of the distribution are in Hertz.
$\boldsymbol{H}_{P}^{(a)}$	Pitch distribution (PD) extracted from the audio fragment (a). The bins of the distribution are in Hertz.
$\boldsymbol{H}_{PC}^{(a)}$	Pitch-class distribution (PCD) extracted from the audio fragment (a). The bins of the distribution are in Hertz.
$h_{n}^{(a)}$	The value to the n^{th} bin of the pitch distribution (PD) or pitch-class distribution (PCD) $\boldsymbol{H}^{(a)}$, extracted from the audio fragment (a).
$h_{P,n}^{(a)}$	The value to the n^{th} bin of the pitch distribution (PD) $\boldsymbol{H}_{P}^{(a)}$, extracted from the audio fragment (a).
$h_{PC,n}^{(a)}$	The value to the n^{th} bin of the pitch-class distribution (PCD) $\boldsymbol{H}_{PC}^{(a)}$, extracted from the audio fragment (a).
$\hat{\boldsymbol{H}}^{x,(a)}$	Pitch distribution (PD) or pitch-class distribution (PCD), extracted from the audio fragment (a). The bins of the distribution are in cents with the 0th bin centered around the reference frequency x.
$\hat{\boldsymbol{H}}_{P}^{x,(a)}$	Pitch distribution (PD) extracted from the audio fragment (a). The bins of the distribution are in cents with the 0th bin centered around the reference frequency x.
$\hat{\boldsymbol{H}}_{PC}^{x,(a)}$	Pitch-class distribution (PCD) extracted from the audio fragment (a). The bins of the distribution are in cents with the 0th bin centered around the reference frequency x.

Symbol	Description
$\hat{h}_n^{x,(a)}$	The value to the n^{th} bin of the pitch distribution (PD) or pitch-class distribution (PCD) $\hat{\boldsymbol{H}}^{x,(a)}$, extracted from the audio fragment (a) with the distribution having its 0^{th} bin centered around the reference frequency x.
$\hat{h}_{P,n}^{x,(a)}$	The value to the n^{th} bin of the pitch distribution (PD) $\hat{\boldsymbol{H}}_P^{x,(a)}$, extracted from the audio fragment (a) with the distribution having its 0^{th} bin centered around the reference frequency x.
$\hat{h}_{PC,n}^{x,(a)}$	The value to the n^{th} bin of the pitch-class distribution (PCD) $\hat{\boldsymbol{H}}_{PC}^{x,(a)}$, extracted from the audio fragment (a) with the distribution having its 0^{th} bin centered around the reference frequency x.
ℓ_n	Accumulator function of the n^{th} bin of a pitch distribution (PD) or pitch-class distribution (PCD).
$\ell_{P,n}$	Accumulator function of the n^{th} bin of a pitch distribution (PD).
$\ell_{PC,n}$	Accumulator function of the n^{th} bin of a pitch-class distribution (PCD).
$b\left(\hat{\boldsymbol{H}}\right)$	Bin size of the pitch distribution (PD) or pitch-class distribution (PCD).
$\sigma\left(\hat{\boldsymbol{H}}\right)$	The width in the standard deviations of the Gaussian kernel used to “smooth” the pitch distribution (PD) or pitch-class distribution (PCD).
$\mu^{(a)}$	(Estimated) makam of an audio fragment (a).
$\mathfrak{m}^{(a)}$	“True” makam of an audio fragment (a).
$\mathcal{M}$	The set of all makams.
$\kappa^{(a)}$	Tonic pitch or pitch-class of an audio fragment (a).
$\mathfrak{k}^{(a)}$	“True” tonic pitch or pitch-class of an audio fragment (a).
$\boldsymbol{\Phi}^{(a)}$	Set of stable pitches or pitch classes in an audio fragment (a).

Symbol	Description
$\phi_i^{(a)}$	A stable pitch or pitch class in an audio fragment (a).
$\mathbf{\Phi}_P^{(a)}$	Set of stable pitches in an audio fragment (a).
$\phi_{P,i}^{(a)}$	A stable pitch in an audio fragment (a).
$\mathbf{\Phi}_{PC}^{(a)}$	Set of stable pitches in an audio fragment (a).
$\phi_{PC,i}^{(a)}$	A stable pitch in an audio fragment (a).
$\delta(\boldsymbol{H})$	Minimum ratio between the value of a peak and the highest value in a pitch distribution (PD) or pitch-class distribution (PCD), for the peak to be selected as a stable pitch or pitch-class.
$\boldsymbol{\mathcal{T}}$	The training model obtained for tonic identification and/or makam recognition using the either of the training schemes explained in (Gedik & Bozkurt, 2010) or (Chordia & Şentürk, 2013).
$\diamondsuit(\boldsymbol{x}, \boldsymbol{y})$	Distance or dissimilarity between two pitch distributions (PDs) or pitch-class distributions (PCDs) $\boldsymbol{x}$ and $\boldsymbol{y}$.
$\hat{\mathbf{\Phi}}_P^{\kappa^{(a)},(a)}$	Set of performed scale degrees in an audio fragment (a).
$\hat{\phi}_{P,i}^{\kappa^{(a)},(a)}$	A perfomed scale degree in an audio fragment (a).
	Joint Audio and Score Analysis
$\bar{\mathfrak{S}}^{(a)}$	“True” audio section sequence in an audio fragment (a).
$\mathfrak{S}^{(a)}$	“True” audio section symbol sequence in an audio fragment (a).
$\bar{\mathfrak{s}}_i^{(a)}$	i^{th} “true” section in the $\bar{\mathfrak{s}}_i^{(a)}$.
$\mathfrak{s}_i^{(a)}$	i^{th} “true” section symbol in the $\mathfrak{S}^{(a)}$.
$\bar{\mathfrak{N}}^{\left(\bar{\mathfrak{s}}_i^{(a)}\right)}$	“True” audio note sequence in the “true” section $\bar{\mathfrak{s}}_i^{(a)}$.
$\bar{\mathfrak{n}}_k^{\left(\bar{\mathfrak{s}}_i^{(a)}\right)}$	k^{th} “true” note in the i^{th} “true” section of the $\bar{\mathfrak{s}}_i^{(a)}$.

Symbol	Description
$\mathfrak{N}^{\left(\bar{\mathfrak{s}}_i^{(a)}\right)}$	"True" audio note symbol sequence in the "true" section $\bar{\mathfrak{s}}_i^{(a)}$.
$\mathfrak{n}_k^{\left(\bar{\mathfrak{s}}_i^{(a)}\right)}$	k^{th} "true" note symbol in the i^{th} "true" section of the $\bar{\mathfrak{s}}_i^{(a)}$.
$\tau^{(x)}$	The tempo of an audio recording or a music score fragment (x) in beats per minute (bpm).
$\hat{\tau}^{b,(x)}$	The tempo of an audio recording or a music score fragment (x) in bpm relative to the nominal tempo indicated in the relevant music score fragment (b).
$\boldsymbol{D}^{x,\left(a,\bar{f}^{(b)}\right)}$	Distance matrix between the audio recording (a) and the score fragment $(\bar{f}^{(b)})$. The feature extracted from (a) (predominant melody or HPCPs) is normalized with respect to the reference pitch or pitch class x.
$\boldsymbol{B}^{x,\left(a,\bar{f}^{(b)}\right)}$	Binary similarity matrix between the audio recording (a) and the score fragment $(\bar{f}^{(b)})$. The feature extracted from (a) (predominant melody or HPCPs) is normalized with respect to the reference pitch or pitch class x.
$\beta(\boldsymbol{B})$	Binarization threshold used to convert a distance matrix to a binary similarity matrix.
$\boldsymbol{A}^{x,\left(a,\bar{f}^{(b)}\right)}$	Accumulated cost matrix between the audio recording (a) and the score fragment $(\bar{f}^{(b)})$. The feature extracted from (a) (predominant melody or HPCPs) is normalized with respect to the reference pitch or pitch class x.
$\pi^{x}\left(\bar{f}^{(a)},\bar{f}^{(b)}\right)$	A link estimated between the audio fragment $(\bar{f}^{(a)})$ and the score fragment $(\bar{f}^{(b)})$. The feature extracted from (a) (predominant melody or HPCPs) is normalized with respect to the reference pitch or pitch class x. Here, the fragments have the same label, i.e. $f^{(a)} = f^{(b)}$.

Symbol	Description
$\Pi^x(a, b, y)$	The set of links between the fragments in the audio recording (a) and the music score (b) with the label y. The feature extracted from (a) (predominant melody or HPCPs) is normalized with respect to the reference pitch or pitch class x.
$\boldsymbol{\varpi}^x\left(\bar{f}^{(a)}, \bar{f}^{(b)}\right)$	Path followed by the link, $\pi^x\left(\bar{f}^{(a)}, \bar{f}^{(b)}\right)$, estimated between the audio fragment $(\bar{f}^{(a)})$ and the score fragment $(\bar{f}^{(b)})$. The feature extracted from (a) (predominant melody or HPCPs) is normalized with respect to the reference pitch or pitch class x.
$\varpi_i^x\left(\bar{f}^{(a)}, \bar{f}^{(b)}\right)$	i^{th} point in the path followed by the link, $\pi^x\left(\bar{f}^{(a)}, \bar{f}^{(b)}\right)$, estimated between the audio fragment $(\bar{f}^{(a)})$ and the score fragment $(\bar{f}^{(b)})$. The feature extracted from (a) (predominant melody or HPCPs) is normalized with respect to the reference pitch or pitch class x.
$\nu^x\left(\bar{f}^{(a)}, \bar{f}^{(b)}\right)$	The similarity between the audio fragment $(\bar{f}^{(a)})$ and the score fragment $(\bar{f}^{(b)})$. The feature extracted from (a) (predominant melody or HPCPs) is normalized with respect to the reference pitch or pitch class x.
$\nu\left(\phi_{PC,i}^{(a)}\right)$	Weight of the i^{th} stable pitch class $\phi_{PC,i}^{(a)}$ in score-informed tonic identification applied to the audio recording (a).
N_{max}	Maximum order of a variable-length Markov model (VLMM).
$\xi_{i \to j}$	Transition probability associated with an edge $e_{i \to j}$ from the i^{th} to the j^{th} node.
$\nu(\mathbf{p})$	Total weight of a path $\mathbf{p}$.

1

List of Figures

List of Tables

Chapter 1

Introduction

Automatic content analysis and description have been two of the most instrumental areas of information technology. From spam filters to automatic face recognition, the applications of these areas have been showing an incredible progress in the last decade in terms of performance, usability and accessibility. Recently, analysis of large-scale data has also opened up new horizons for acquisition and *en masse* summarization of relevant content along with relational discovery of studied corpora in a wide variety of use cases. These technologies have been a major stimuli on shaping how we interact, manipulate and exploit information in our daily lives, and consequently influencing the contemporary human society.

Content analysis methodologies typically employ data-driven approaches designed with the type of information source, the addressed task and the content itself in mind. In this regard, music content analysis presents numerous challenges brought by the properties (e.g. melody, rhythm and structure) of the studied music culture in various domains such as audio recordings, music scores and textual information. Furthermore each music style has its peculiar musical characteristics, which has the interpretations inclusive to its own cultural context. Therefore, any technology that attempts to extract, process, describe and discover the musical content has to be designed with an awareness on the culture-specific properties of the studied music tradition in order to obtain a coherent, musically meaningful and culturally relevant information.

Music is a complex phenomenon and there are many types of data sources that can be used to study it, such as audio recordings, scores, videos, lyrics and social tags. At the same time, for a given piece there might be many versions for each type of data, for example we find cover songs, various orchestrations and diverse lyrics in multiple languages. Each type of data source offers different ways to study, experience and appreciate music. If the different information sources of a given piece are linked with each other (Thomas, Fremerey, Müller, & Clausen, 2012), we can take advantage of their complementary aspects to study musical phenomena that might be hard or impossible to investigate if we have to study the various data sources separately.

The linking of the different information sources can be done at different time spans, e.g. linking entire documents (Ellis & Poliner, 2007; Martin, Robine, & Hanna, 2009; Serrà, Serra, & Andrzejak, 2009), structural elements (Müller & Ewert, 2008), musical phrases (Wang, 2003; Pikrakis, Theodoridis, & Kamarotos, 2003), or at note/phoneme level (Niedermayer, 2012; Fujihara & Goto, 2012). Moreover there might be substantial differences between the information sources (even among the ones of the same type) such as the format of the data, level of detail and genre/culture-specific characteristics. Thus, we need content-based (Casey et al., 2008), application-specific and knowledge-driven methodologies to obtain meaningful features and relationships between the information sources. The current state of the art in music information research (MIR) is mainly focused on Eurogenetic[1] styles of music (Tzanetakis, Kapur, Schloss, & Wright, 2007) and we need to develop methodologies that incorporate culture-related knowledge to understand and analyze the characteristics of other musical traditions (Holzapfel, 2010; Şentürk, 2011; Serra, 2011).

The thesis proposes several computational approaches to automatically analyze and describe music scores and audio recordings of Ottoman-Turkish makam music (OTMM). The main contributions of the thesis are the music corpus that has been created to carry out the research and the audio-score alignment methodology developed for the analysis of the corpus. In addition, several novel computational analysis methodologies are presented in the context of

[1] We apply this term because we want to avoid the misleading dichotomy of Western and non-Western music.

common MIR tasks of relevance for OTMM. Some example tasks are predominant melody extraction, tonic identification, tempo estimation, makam recognition, tuning analysis, structural analysis and melodic progression analysis.

The goals of the thesis are:

- Create a research corpus representative of the studied aspects of OTMM

- Develop novel audio-score alignment based analysis methodologies, which address the culture-specific challenges posed by the musical characteristics of OTMM

- Integrate the existing state-of-the-art in music score analysis and audio analysis algorithms with the implementations of the developed methodologies for the automatic automatic description and discovery of large-scale corpora, consisting of music scores and audio recordings.

1.1 Outline

The thesis is organized into seven chapters, wherein the main contributions are contained in Chapters 3–6.

Chapter 2 gives a review of their musical and scientifica background relevant to the thesis work. The Chapter starts with a brief introduction to OTMM and identifies several computational challenges brought by its musical characteristics. Next, the state-of-the-art in automatic description and audio-score alignment. For organizational purposes, the rest of the state-of-the-art is discussed in relevant sections, in which each computational task is introduced.

Chapter 3 is about the CompMusic OTMM corpus created in the scope of the CompMusic project in collaboration with several other researchers in the project. The Chapter presents the statistics of the music score collection, audio collection, metadata etc. It also validates the corpus according to some criteria such as *completeness*, *coverage* and *quality*. Then, the test datasets, which are drawn out of the CompMusic OTMM corpus to evaluate specific computational methodologies, are showcased. The Chapter also includes a brief explanation of progress in the creation of the makam ontologies, which has started as part of this thesis.

Chapter 4 explains the developed score analysis methodologies. It starts by describing how the metadata related to music scores are parsed. Next the extracted melodic and lyrics features are introduced. Using these features, structural organization of the music scores are analyzed. The proposed structure analysis methodology presents a novel semiotic labeling scheme based on the melodic and lyrics similarities between the sections and the phrases. From the metadata and the structural analysis, an automatic description of the score collection is obtained. The Chapter also highlights the automatic score converters and validators developed for the music collection.

Chapter 5 makes an overview of the melody-related audio analysis methodologies applied to OTMM. It proposes several generalizations and improvements on the existing state-of-the-art, specifically in predominant melody extraction and joint makam-tonic estimation. The developed analysis algorithms are used to obtain an automatic description of the whole audio collection.

Chapter 6 presents the core contribution of the thesis, which is the joint audio-score analysis. The joint analysis is based on a novel audio-score alignment, which is robust to many performance aspects of OTMM such as tonic transpositions, tempo variations, tuning and intonation deviations, non-notated embellishments and heterophony. The automatic description obtained from joint analysis links the audio and the symbolic data. Moreover, the resultant score-informed analysis improves the performance compared with results obtained from the state-of-the-art methods solely relying on audio data.

Chapter 7 showcases the web application developed for the discovery of CompMusic OTMM corpus, presents an overall summary of the thesis and discussfuture directions, which could be undertaken.

There are several Appendices in the thesis. Appendix A describes the preliminary experiments in section-level audio score alignment. Appendix B presents the applications of the developed methodologies on different music cultures. Appendix C the contributions to open and reproducible research Throughout the thesis. Appendix D provides a mirror of content in the companion web page. Appendix E lists my publications. The glossary for the abbreviations and terms used throughout the thesis is in Appendix F.

Chapter 2

Background

2.1 Ottoman-Turkish Makam Music

In a large geographical region of Asia, north Africa and east Europe, there are numerous music traditions described around the concepts "makam/maqam/mugam" etc., which share similar practice and terminology. The thesis focuses on the makam music tradition, which proliferated in the Ottoman Empire and continues its legacy principally in Turkey (Tanrıkorur, 2011; Behar, 2015). I term the music as Ottoman-Turkish makam music (OTMM) throughout the text for consistency's sake. This name is derived from "Osmanlı-Türk Musikisi" (Ottoman-Turkish Music) coined by Behar (2015) with an added emphasis on makam, the melodic structure of this music culture.

The musical terminology presented in this Section is mainly use to describe the classical/art ("klasik/sanat" in Turkish) repertoire. Nevertheless, the concepts (albeit, with slightly different wording) also hold for folk and some popular music (Signell, 1986; Tanrıkorur, 2011), which are also represented in the studied corpus (Section 3.1). For a more indepth review of OTMM in the context of computational studies, the readers are referred to (Bozkurt, Ayangil, & Holzapfel, 2014).

2.1.1 Performance Practice

OTMM has been predominantly an oral tradition for centuries. For this reason, the performance practice is the fundamental unit of OTMM (Ederer, 2011). In many cases, the compositions have been modified through oral propogation to a degree such that the original musical intend may have been diminished (Behar, 2015). In paralel, some pieces have numerous versions, which are a consequence of the divergences between transmissions from different masters.

The performers typically perform simultaneous variations of the same melody in their own register, a phenomenon commonly referred to as heterophony (Cooke, accessed April 5, 2013). They are supposed to show their virtuosity by adding embellishments, inserting/repeating/omitting notes, altering timings, and changing the tuning and intonation. The intonation of some intervals in a performance might differ from the "theoretical" intervals as much as a semi-tone (Signell, 1986). There is typically no lead instrument in instrumental performances. Vocals typically lead the melody; nonetheless heterophony is retained.

The musicians are flexible in playing with the structural organization. They might insert phrase or section repetitions, insertions or omissions. In addition, they may improvise before, after and within the performance of a composition.

There is no definite reference frequency (e.g. $A4 = 440$Hz) to tune the performance tonic. Moreover, there are a number of different transpositions (ahenk in Turkish), any of which might be favored over others due to instrument/vocal range or aesthetic concerns (Ederer, 2011).

Due to the expressive decisions explained above, there may be high degrees of variance between different interpretations of the same piece.

2.1.2 Theory and Notation

There are several theories attempting to explain the makam practice (Karadeniz, 1984; Özkan, 2006; Yarman, 2008). The mainstream theory is AEU theory (Arel, 1968). AEU theory argues that there are 24 equal intervals. A whole tone is divided into 9 equidistant intervals, each of which is termed as a Holderian

Name	Flat	Sharp	Hc
Koma (en. Comma)	𝄳	𝄲	1
Bakiye	𝄬	♯	4
Küçük mücennep	♭	𝄰	5
Büyük mücennep	𝄭	𝄱	8

Table 2.1: The accidental symbols defined in extended Western notation used in OTMM and their theoretical intervals in Hc according to the AEU theory.

comma (Hc) (Ederer, 2011). Tura (1988) states that these intervals can be approximated from 53-tone-equal-tempered (TET) intervals (i.e. 1 Hc $= \frac{1200}{53} \approx 22.64$ cents). Bozkurt, Yarman, Karaosmanoğlu, and Akkoç (2009) analyzed several performances of renowned musicians to assess the tunings in different makams, and showed that the current music theories are not able to explain the intervallic relations well.

Since early 20^{th} century, a score representation extending the traditional Western music notation has been used as a complement to the oral practice (Popescu-Judetz, 1996; Ayangil, 2008). The extended Western notation typically follows the rules of AEU theory. Table 2.1 lists the accidental symbols specific to OTMM defined in this notation.

The music scores tend to notate simple melodic lines and they do not indicate the heterophonic interactions. Most of the scores are transcriptions, and they are written sometimes centuries after the pieces were composed. As a result, it is common to observe several music scores for a certain composition, each of which depicts an interpretation of the composition within the oral tradition. The musicians may follow a particular music score as a guideline. Nevertheless, they considerably extend the notated "musical idea" during the performance as described in Section 2.1.1.

2.1.3 Melodic, Rhythmic, and Formal Structure

Most of the melodic aspects of OTMM can be explained by the term makam. Makams constitute the melodic structure of most of the traditional music repertoires in Turkey. Makams are modal structures, where the melodies typically revolve around an initial

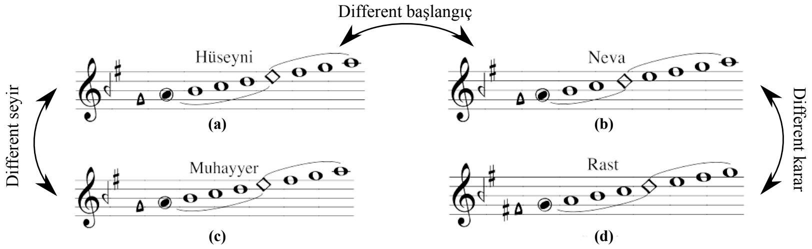


Figure 2.1: The scales of four makams sharing the same key signature: **a)** Hüseyni, **b)** Neva, **c)** Muhayyer, **d)** Rast. The diamond shaped note, the filled note and the triangle shaped note depict the initial tone (başlangıç), final tone (karar) and the seventh (yeden), respectively.

tone ("başlangıç" or "güçlü" in Turkish) and a final tone (karar in Turkish) (Ederer, 2011; Bozkurt, Ayangil, & Holzapfel, 2014). The karar is typically used synonymous to tonic. Each makam has a particular scale, which gives the "lifeless" skeleton of the makam (Signell, 1986). A makam is gains its character through its melodic progression (seyir in Turkish) (Tanrıkorur, 2011). Ayangil (2001) describes makam as "descriptions of seyir rules as road maps" (Bozkurt, Ayangil, & Holzapfel, 2014).

Figure 2.1 shows the scales of four makams, which share the same key signature. In addition, Hüseyni, Neva and Muhayyer have the same scale. Hüseyni, Neva and Rast may be distinguished from the karar and güçlü notes. On the other hand, Hüseyni and Muhayyer makams only differs from each other with their seyirs, which are theoretically explained as ascending-descending and descending, respectively. Bozkurt (2015) conducted a computational analysis of seyir (the procedure will be described in Section 5.10), and discussed that the computed seyir feature can be used to discriminate makams.

The metric structure is explained by usul. A certain usul roughly defines the cyclic meter, and it can be described by a group of strokes with different velocities, which imply the beats and downbeats in the rhythmic pattern (Marcus, 2001). Nevertheless, usul is a wider concept, which is not limited to metric implications, since a change in usul can disrupt the seyir, and even change the perception of the makam (Tura, 1988). The number of pulses ("zaman" in Turkish) in an usul cycle might vary from 2 up to 120. An usul might have different variants with respect to tempo, which are called mertebes.[1]

The overall structural organization of OTMM is described by form. Each form is described by its distinct structural characteristics. For example, peşrev, sazsemaisi (the two most common instrumental forms in the classical repertoire) commonly consists of four distinct hanes and a teslim section, which typically follow a *verse-refrain*-like structure. Nevertheless, there are peşrevs, which have no teslim, in which case the second half of each hane strongly resembles each other (Karadeniz, 1984). The 4th hane in the sazsemaisi form is usually longer, includes rhythmic changes, and it might be divided into smaller substructures. Each of these substructures might have a different tempo with respect to the overall tempo of the piece. Similarly, a şarkı (the most common vocal form in the classical repertoire) is typically divided into sections called aranağme, zemin, nakarat and meyan. The typical order of the sections is aranağme, zemin, nakarat, meyan and nakarat. Except of the instrumental introduction aranağme, all the sections are vocal and determined by the lines of the lyrics. Each line in the lyrics is usually repeated, but the melody in the repetition might be different. Some şarkıs have a gazel section (vocal improvisation), for which the lyrics are provided in the score, without any melody.

The forms may be also classified into several categories such as compositional/improvisational, classical/folk, vocal/instrumental, religious and military. Nonetheless, the categorization should not be considered as an absolute, as there are many transitive examples (Tanrıkorur, 2011; Behar, 2015).

2.2 Computational Challenges

Bozkurt, Ayangil, and Holzapfel (2014) discussed the computational challenges brought by the musical characteristics of OTMM. Below, I recapitulate some of these challenges, which are addressed throughout the thesis:

[1] A consice explanation of usul with audio examples are given in the CompMusic website: `http://compmusic.upf.edu/examples-usul-mmt`.

- Melody extraction algorithms might not perform well in recordings of OTMM due to the heterophonic interactions (Section 5.2).
- Tonic identification is necessary for almost all types of melodic analysis (Sections 5.7 and 6.4).
- Since OTMM is inherently microtonal, many computational analysis steps (e.g. Sections 5.2, 5.7 and 6.8) need to provide a high pitch precision.
- A performance might be realized by taking another performance (e.g. of an indisputable master) or another music score as the reference; or it may be a genuinely "original" interpretation.[2] Therefore, the audio-score alignment methodologies should be able to find inexact matches between the information sources (e.g. in score-informed composition identification described in Section 6.6).
- Even if we know that the performance takes the music score as the reference, audio-score alignment method has to incorporate a flexible matching scheme due to the interpretative freedom in the performances and the simplicity of the music notation (Chapter 6).
- In addition, audio-score alignment methodologies should be designed to handle structural differences between music scores and audio performances (Section 6.7).[3]

2.3 Relevant Works

There has been an increasing interest in (comparative) computational studies on many musical cultures across the world such as Indian Art Musics (IAM), European folk music, Persian music and African musics (Tzanetakis et al., 2007; Moelants et al., 2007; Tzanetakis, 2014). In parallel, most of the academic studies on the com-

[2] Take *Kudsi Erguner Ensemble*'s interpretation of Tanburi Cemil Bey's composition, *Şedaraban Peşrevi* as an example of original interpretation: `http://dunya.compmusic.upf.edu/makam/recording/97be5bdd-cef0-4103-bbb7-bff77d6b0a30`

[3] In *Neva* and *İhsan Özgen*'s performance of *Uşşak Sazsemaisi* (`http://dunya.compmusic.upf.edu/makam/recording/0756f4f9-7fe2-48a8-b1c8-47ef8be9377f`), the performers repeat the fourth hane twice and also repeat the teslim twice after the third and the fourth hane. In addition *Neva Özgen* performs a taksim between the fourth hanes.

putational analysis of OTMM has also been realized in the last decade. Bozkurt, Ayangil, and Holzapfel (2014) presents a comprehensive survey of computational studies on OTMM. To avoid repetition, this Section discusses previous works relevant to the thesis, which are either outside the context of OTMM or was not addressed in (Bozkurt, Ayangil, & Holzapfel, 2014) extensively.

2.3.1 Research and Test Corpora

As Serra (2014) discussed, one of the most important facets of music information research is the creation, organization and usage of music corpora representative of the studied musical phenomenon. Such corpora can be utilized further to create test corpora (datasets), which are aimed for the evaluation of computational methodologies proposed for specific tasks. There are numerous test corpora, which have been used extensively in MIR such as GTZAN Genre Collection (Tzanetakis & Cook, 2002), RWC dataset (Goto, Hashiguchi, Nishimura, & Oka, 2003), Mazurka Project[4] and Music-Net (Thickstun, Harchaoui, & Kakade, 2016). These datasets are typically annotated and/or curated by music experts. The creation of these datasets are time-consuming and require extensive music knowledge, therefore these types of corpora are typically small to medium in size.

Currently, most of the music corpora belong to the Eurogenetic music genres/cultures. One of the biggest aims of the CompMusic project by creating large and high-quality music corpora for the studied music traditions (Serra, 2014). Currently, CompMusic hosts the largest corpora of OTMM (Uyar, Atlı, Şentürk, Bozkurt, & Serra, 2014) (explained in Chapter 3 in detail), IAM (Srinivasamurthy, Koduri, Gulati, Ishwar, & Serra, 2014), Beijing opera (Rafael & Serra, 2014), Arab-Andalucian music (Sordo, Chaachoo, & Serra, 2014). Other relevant music corpora include the Meertens Dutch folk song collection (van Kranenburg, Janssen, & Volk, 2016) and the COFLA flamenco corpus (Kroher, Díaz-Báñez, Mora, & Gómez, 2016).

Currently, Million Song dataset (Bertin-Mahieux, Ellis, Whitman, & Lamere, 2011) and AcousticBrainz (Porter, Bogdanov, Kaye, Tsukanov, & Serra, 2015) are the largest audio recording reposito-

[4] http://www.mazurka.org.uk/

ries for MIR. The data in both repositories are linked with other relevant information sources such as MusicBrainz, 7digital.[5] In both cases, only the automatic description of the audio recordings are due to copyright reasons. Due to computational and storage reasons, both Million Song dataset and AcousticBrainz provide either limited temporal information on the analyzed tracks. Moreover, they do not represent many music genres/traditions adequately.

2.3.2 Audio-Score Alignment

Audio-score alignment aims to synchronize the musical events in the score of a composition with the corresponding (generally latent) events in the audio recording of the same composition. The alignment can be applied to in different granularities (Thomas et al., 2012) such as notes (Cont, 2010; Niedermayer, 2012), measures (Fremerey, Müller, & Clausen, 2010), phrases (Nakamura, Nakamura, & Sagayama, 2016) or sections (Şentürk, Holzapfel, & Serra, 2014; Holzapfel, Şimşekli, Şentürk, & Cemgil, 2015). Generally, if score and audio recording of a piece are aligned on the note or measure level, section borders in the audio can be inferred from the time stamps of the linked notes/measures in the score and audio (Thomas et al., 2012). Likewise, a method might output a fine-grained result when the aim is aligning higher-level structures (Holzapfel et al., 2015).

As a result of the alignment, the time-series data in the symbolic and audio domains are linked with each other (Thomas et al., 2012). The linked data could facilitate many additional computational tasks such as automatic accompaniment (Cont, 2010; Arzt, Böck, & Widmer, 2012), music discovery (Şentürk, Ferraro, Porter, & Serra, 2015) and computational musicology (Devaney, Mandel, & Fujinaga, 2012; Abesser, Frieler, Cano, Pfleiderer, & Zaddach, 2016). Audio-score alignment might simplify several tasks such as audio transcription (Benetos & Holzapfel, 2015), meter tracking (Srinivasamurthy, Holzapfel, Cemgil, & Serra, 2016) and structural analysis (Paulus, Müller, & Klapuri, 2010), which would require sophisticated audio analysis techniques.

[5] https://www.7digital.com/

The current state-of-the-art on audio-score alignment follows two main approaches: dynamic time warping (DTW) (Dixon & Widmer, 2005; Niedermayer, 2012; Rodriguez-Serrano, Carabias-Orti, Vera-Candeas, & Martinez-Munoz, 2016; Prätzlich, Driedger, & Müller, 2016) and Hidden Markov model (HMM) (Peeling, Cemgil, & Godsill, 2007; Cont, 2010; Maezawa, Itoyama, Yoshii, & Okuno, 2014; Nakamura et al., 2016), though there also exists methods based on Conditional Random Fields (CRFs) (Joder, Essid, & Member, 2010), audio fingerprinting (Arzt, Widmer, & Sonnleitner, 2014), and (deep) neural networks (Dorfer, Arzt, & Widmer, 2016). Typically the alignment methodologies are specialized to address the specific task by enhancing/adapting the basic method according to the properties of the studied music corpus (Fremerey et al., 2010; Devaney et al., 2012; Rodriguez-Serrano et al., 2016).

Table 2.2 lists numerous audio-to-audio and audio-score alignment methodologies, which are relevant to the in terms of the computational problems and challenges (Section 2.2. **Str.**, **Tra.**, **Mic.** and **Exp**, indicate if the proposed method is designed to address structural differences, transpositions, microtonality and "expressive" interpretations, respectively. The green and yellow colors in the **Str.** column illustrate whether the alignment handles all structural insertions, repetitions, deletions, and unrelated event (e.g. improvisation, speech) additions, or not. The rest of this Section will mainly discuss these methodologies. Note that the last two rows of Table 2.2 show the alignment methodologies partially developed in the context of the thesis for comparison. These methodologies will be explained more in detail in (Chapter 6).

Structure

In general, approaches of audio-score alignment assumes that the score and the target audio recording are structurally identical, i.e. there are no phrase repetitions and omissions in the performance. (Fremerey et al., 2010) extended the classical DTW and introduced JumpDTW, which is able to handle such structural non-linearities. However, due to the its level of granularity, audio-score alignment is computationally expensive.

Since section linking is aimed at linking score and audio recordings on the level of structural elements, it is closely related to audio

Table 2.2: An overview of relevant alignment methodologies.

Name	Method	Music Corpus	Feature	Target / Setting	Query	Str.	Tra.	Mic.	Exp.
(Tekin, Anagnostopoulou, & Tomita, 2005)	Linked Lists	Western Classical	Symbolic	Online Piano MIDI Output	Complete Music Score	✓		N/A	✓
(Pardo & Birmingham, 2005)	HMM	Jazz	Symbolic	Complete (Synthesized) Audio Recording	Complete Music Score	✓			
(Müller & Appelt, 2008)	Dynamic Programming	Various Eurogenetic	Chroma	Complete Audio Recording	Complete Music Score	✓			
(Müller, Grosche, & Wiering, 2009)	iterative subsequence dynamic time warping (ISDTW)	Dutch Folk Songs	Chroma	Complete Audio Recording	Complete Music Score	✓	✓	✓	✓
(Fremerey et al., 2010)	JumpDTW	Western Classical	Chroma	Complete Audio Recording	Complete Music Score	✓			
(Joder et al., 2010)	Conditional Random Fields	Western Classical	Chroma, Onset, Tempogram	Complete (Synthesized) Audio Recording	Complete Music Score	✓	✓		
(Arzt & Widmer, 2010)	Rough Position Estimator, Multiple Online DTWs, Descision Maker	Western Classical	FFT-based	Online Audio Recording	Complete Music Score	✓			
(Niedermayer, 2009)	Multiscale DTW	Western Classical	Chroma	Complete Audio Recording	Complete Music Score				
(Duan & Pardo, 2011)	Dynamic Programming	Jazz	Chroma	Complete Audio Recording	Complete Music Score	✓	✓		✓
(Arzt et al., 2012, 2014)	Audio Transcription, Fingerprinting, Database Querying	Western Classical	Transcribed Audio, Symbolic Score	Online Audio Recording	Complete Music Score	✓	✓		✓
(Devaney et al., 2012)	DTW, HMM	Western Classical	YIN + onset	Complete Audio Recording	Complete Music Score			✓	✓
(Grachten, Gasser, Arzt, & Widmer, 2013)	Needleman-Wunsch time warping	Western Classical	MFCC, Constant Q transform (CQT), PST or LNSO/NC	Complete Audio Recording	Complete Music Score	✓			
(Prätzlich & Müller, 2014)	DTW-based	Western Classical	Chroma	Complete Audio Recording	Sections in the Audio Recording	✓			
(Nakamura et al., 2016)	HMM	Western Classical	CQT	Online Audio Recording	Complete Music Score	✓			✓
(Şentürk, Holzapfel, & Serra, 2014; Şentürk, Gulati, & Serra, 2014)	Hough transform, SDTW, graph analysis	OTMM	Predominant Melody	Complete Audio Recording	Sections in the Music Score	✓	✓	✓	✓
(Holzapfel et al., 2015)	HHMM	OTMM	Predominant Melody	Complete Audio Recording	Complete Music Score	✓	✓	✓	✓

structure analysis (Paulus et al., 2010). The state of the art methods on structure analysis are mostly aimed at segmenting audio recordings of popular Eurogenetic music into repeating and mutually exclusive sections. For such segmentation tasks, self-similarity analysis (Cooper & Foote, 2002; Goto, 2003) is typically employed. These methods first compute a series of frame-based audio features from the signal. Then all mutual similarities between the features are calculated and stored in a so-called self similarity matrix, where each element describes the mutual similarity between the temporal frames. In the resulting square matrix, repetitions cause parallel lines to the diagonal with 45 degrees and rectangular patterns in the similarity matrix. This directional constraint makes it possible to identify the repetitions and 2-D sub-patterns inside the matrix.

Since the sections in a composition follow a certain sequential order, the extracted information can be formulated as a directed acyclic graph (DAG) (Newman, 2010). (Paulus & Klapuri, 2009) use this concept in self-similarity analysis. They generate a number of border candidates for the sections in the audio recording and create a DAG from all possible border candidates. Then, they use a greedy search algorithm to divide the audio recording into sections.

Transposition

Müller and Clausen (2007) introduced transposition-independent similarity matrices. This matrix computed from chroma features extracted from two audio recordings (real or synthetic) to be compared. In the point-to-point distance computation (using cosine distance) one of the frames is shifted circularly. The distance is assigned as the minimum of the computed distances. Müller et al. (2009) later uses the transposition-independent similarity matrices in analyzing Dutch folk songs to compansate the tonic deviations within a recording.

Expressivity

When fragments of audio or score are to be linked, the angle of the diagonal lines in the similarity matrix computed are not 45 degrees, unless the tempi of both information sources are exactly the same. This problem also occurs in cover song identification (Ellis

& Poliner, 2007; Serrà et al., 2009) for which a similarity matrix is computed using temporal features obtained from a cover song candidate and the original recording. If the similarity matrix is found to have some strong regularities, they are deemed as two different versions of the same piece of music. A proposed solution is to “squarize” the similarity matrix by computing some hypothesis about the tempo difference (Ellis & Poliner, 2007). However, tempo analysis in makam musics is not a straightforward task (Holzapfel & Stylianou, 2009). The sections may also be found by traversing the similarity matrices using dynamic programming (Serrà et al., 2009). On the other hand, dynamic programming is a computationally demanding task.

Chapter 3

Music Corpus and Test Datasets

For computational studies on specific type of musics, there is a need for corpora, which constitutes the studied aspects of the particular music culture. A music corpus may consist of multiple information sources such as audio recordings, music scores, lyrics and editorial metadata. The information sources in the corpus may be also analysed to obtain a description, which extends the corpus itself. The corpora may be grouped into two types: research corpus and test dataset (Serra, 2014). A research corpus is a data collection that represents the ”real world” for a specific research problem. A test dataset is a collection for a specific research task to test, calibrate and evaluate particular methodologies.

Serra (2014) provides such criteria for the design of culture specific corpora, which are specified as *purpose, coverage, completeness, quality* and *reusability*. To elaborate, the *purpose* of the corpora should be well-defined to facilitate research tasks. The corpora should be of good *coverage* to represent the music tradition and include metadata with a high degree of *completeness* related to studied aspects of the music. The corpora should attain a certain *quality* and it should be *re-usable* for future research.

In this Chapter,[1] a corpus for computational research of OTMM

[1] This Chapter is partially published in (Uyar et al., 2014). Here, the statistics of the corpus and test datasets are updated.

is presented, which is designed with these considerations in mind. The corpus is described with respect to the information sources that are used to populate it, namely audio recordings, machine-readable music scores, editorial metadata. For each type of data, the *purpose, coverage, completeness, quality* and *reusability* criteria are discussed, when applicable. The automatic description of the corpus, which is considered as part of the corpus, is described in the Chapters 4–6. The test datasets of OTMM, which are gathered in the scope of the CompMusic Project, are also explained.[2]

The rest of the Chapter is as follows: Section 3.1 explains the CompMusic OTMM corpus and the criteria for creating this research corpus. Section 3.2 gives a detailed information about the test datasets. Section 3.3 introduces the ontologies built for OTMM and Section 3.4 wraps up the Chapter with a brief conclusion.

3.1 Music Corpus

In the CompMusic project, we mainly focus on the melodic and the rhythmic characteristics of OTMM. To study these aspects of the music tradition, we have been collecting audio recordings and music scores. From the audio recordings we can extract the characteristics of interpretations of compositions performed by musicians. The music scores, on the other hand, provide an easy-to-access medium to extract the musical elements. We additionally store the editorial metadata about OTMM. The metadata contains information related to the audio recordings and music scores as well as additional information such as the birth date of the artists and relevant web sources about the entities. The metadata also consists of the relationships between each entity so that the connections within the metadata can be exploited to access relevant information in a structured manner. TheCompMusic OTMM corpus also includes of the automatic description of the music scores and audio recordings that are obtained within the scope of this thesis.

For this purpose, a team of more than 15 collaborators, has been working to collect and label all the available data. In this Section, we explain the audio recordings (Section 3.1.1), the music scores

[2] This Chapter is mainly based on the material presented in (Uyar et al., 2014) with updated statistics of the corpus as of August 2016.

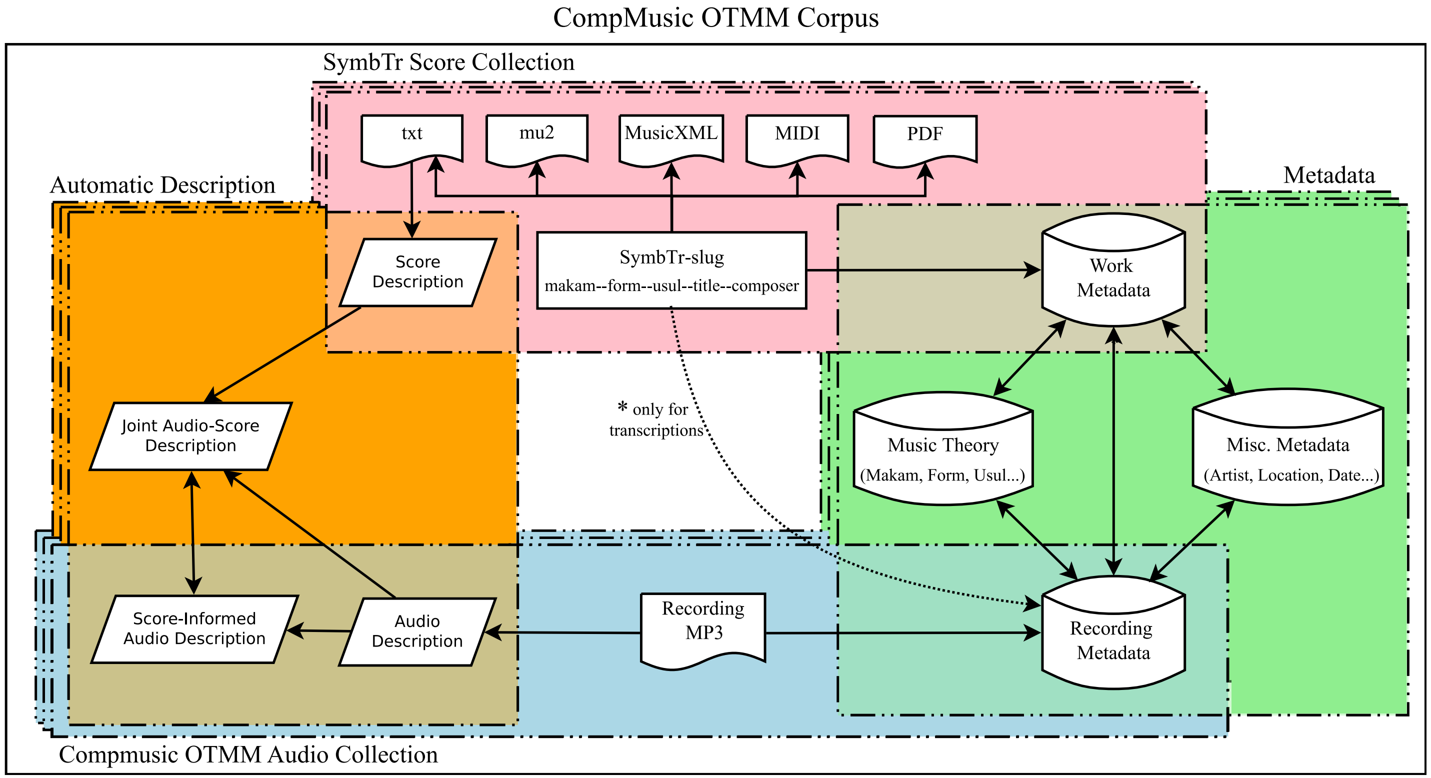


Figure 3.1: The block diagram of the CompMusic OTMM corpus.

(Section 3.1.2) and the editorial metadata (Section 3.1.4) in the research corpus. The metadata related to audio recordings and the music scores are mainly explained within the corresponding source type. The corpus is discussed in terms of the *purpose, coverage, completeness, quality* and *reproducibility* of the audio recordings and the music scores (Serra, 2014). In Section 3.1.4 we mostly focus on the overall statistics of the metadata as well as the statistics of inter-relationships. Section 3.1.5 gives a bried introduction to the automatic description of the corpus.

In our corpus, we use MusicBrainz (Swartz, 2002) to store the metadata. MusicBrainz assigns a unique identifier, called MusicBrainz identifier (MBID) to each entry (e.g. releases, audio recordings, artists).[3] Unless otherwise indicated, all content (except commercial recordings) it the corpus is licensed under the Creative Commons Attribution-NonCommercial 3.0 License (CC BY-NC 3.0) (Spain).

3.1.1 Audio Collection

While creating the corpus, one of our major efforts has been directed to create an audio collection representative of OTMM. The CompMusic OTMM audio collection consists of 6601 stereo recordings. This collection corresponds to more that 420 hours of play time. The collection includes both solo recordings and ensemble/chorus recordings. Note that some recordings exist in multiple releases either identically or with different mastering. Considering such duplicates, there are 6543 unique performances in the audio collection. They span a time period from the start of the 20^{th} century to nowadays. The collection covers various forms, which are part of the classical (e.g. şarkı, sazsemaisi) or folk (e.g. türkü, oyunhavası) music (Table 3.2). Some of the pieces also belong to the religious (e.g. ilahi) or the military (e.g. mehter) repertoire.

Coverage

Historically, TRT has the most representative audio productions of OTMM. However, most of their audio collection is not open

[3] For more information on please refer to `http://musicbrainz.org/doc/MusicBrainz_Identifier`.

	#
Recordings	6601
Works	2928
Artists	811
Releases	356
Makams	111
Usuls	74
Forms	87
Instruments	64
Vocal Types	8

Table 3.1: Number of unique recordings, releases, works, artists, makams, usuls, forms and instruments/voicing in the CompMusic OTMM audio collection.

to public and only a small part of this collection is commercially available. Apart from TRT, there are numerous labels, which have released recordings of OTMM. For these reasons, it is hard to collect the overall statistics of OTMM recordings.

So far, we have focused our efforts on gathering an audio collection of classical repertoire, including the available commercial recordings from TRT and other important labels. We also include several non-commercial recordings, provided that they have a good overall musical and production quality. In Table 3.1, we present the general statistics of the gathered audio collection. Table 3.2 shows the number of the most common makams, forms and usuls in the audio collection.

Completeness

Along with the audio recordings, we also collect editorial metadata given in the album covers. In case an album cover does not provide related metadata (e.g. related work, makam) we attempt to fill the missing metadata by accessing other information sources available. The procedure is as follows: if some of the infomation (e.g. makam, usul, form, composer) is missing, a search with the name of the recording is performed in the online score collections (such as the ones explained in Section 3.1.2 later), and the missing infor-

Makam	#	Form	#	Usul	#
Hicaz	749	Şarkı	2667	Serbest	1425
Nihavent	500	Taksim	1249	Aksak	712
Hüzzam	461	Peşrev	529	Düyek	674
Uşşak	415	Sazsemaisi	473	Aksaksemai	598
Kürdilihicazkar	374	Türkü	232	Kapalı curcuna	421
Rast	372	Yürüksemai	191	Curcuna	362
Hüseyni	345	Beste	190	Sofyan	342
Segah	299	Ağırsemai	166	Yürüksemai	339
Hicazkar	158	İlahi	135	Ağıraksak	216
Mahur	157	Gazel	104	Devr-i kebir	216
Other (100 makams)	2655	(76 forms)	908	(63 usuls)	1888
Total	6485		6844		7194

Table 3.2: The most represented makams, forms and usuls attributes in the audio collection and the corresponding number of instances. Note that multiple compositions and improvisations might be performed in an audio recording. Therefore, an audio recording may have multiple instances of the same type of attribute associated.

mation is obtained from the matched score. For recordings named only after the makam and form such as “Hicaz Peşrev”, since there can be many ”Hicaz Peşrev” compositions, other ”Hicaz Peşrev”s in the audio collection are listened and checked if there exists a match. If a match is found, the corresponding work information is copied.

The completeness of the audio related metadata is shown in Table 3.3. While checking the completeness of the artist metadata in the audio recordings, we assume a recording is complete if it has at least one artist associated with it. Note that this does not imply a strict completeness with respect to the artist metadata since a lot of recordings (esp. ensemble recordings) do not have the complete information about the performers in the album covers.

Quality

The audio recordings are stored in MP3 audio format. This format is chosen due to its quality with a small storage size compared to other audio formats. The audio files are sampled at 44100 Hertz (Hz) and 160 kilobits per second (kbps).

	# Recordings	**% of total**
Releases	6486	98%
Works	5266	80%
Artists	6608	100%
Makams	6163	93%
Forms	6389	97%
Usuls	6090	92%

Table 3.3: The number of audio recordings for which the metadata type is available. 1238 audio recordings are improvisations, which do not have a work. 106 recordings are standalone recordings and hence does not have a release.

In the selection process, the releases in the CompMusic OTMM audio collection are labeled in terms of its "cultural representability." If a release is labeled as "unrepresentative," if it is not of high production quality (except historical recordings) or musical quality, uses non-traditional instruments (e.g. synthesizers, acoustic guitar) extensively or does not strictly belong to a OTMM genre (e.g. contemporary Turkish pop, Arabic maqam). In the audio collection, 46 releases[4] out of 356 releases[5] are labeled as such.

Re-Usability

The non-commercial recordings in our research corpus are freely-available. Most of these non-commercial recordings can be downloaded from *Internet Archive*[6] or the respective websites where they were originally fetched from.[7] The cover arts of most of the releases are available via *CoverArt Archive*.[8]

Due to copyright restrictions, we cannot distribute the commercial audio recordings. On the other hand, they are available for browsing and listening through Dunya-makam (Porter, Sordo, &

[4] Listed in https://musicbrainz.org/collection/9b7a0d92-a756-411d-81da-e855c946f23e

[5] Listed in https://musicbrainz.org/collection/5bfb724f-7e74-45fe-9beb-3e3bdb1a119e

[6] http://tinyurl.com/n9omoue

[7] e.g. http://www.socsci.uci.edu/~rgarfias/films.html

[8] https://coverartarchive.org/

Serra, 2013a; Şentürk et al., 2015) (explained in Section 7.1.1). Moreover the annotations on all the audio recordings and the various features extracted from them are freely distributed under the CC BY-NC 3.0 (Spain). They can be used for computational research purposes and redistributed according to the terms of the licence.

3.1.2 Score-Collection

The existing music scores of OTMM are mostly in physical formats, such as hand-written scores and books. There are also non-machine-readable scores available in digital formats like JPEG and PDF. Typically, these types of scores are not very useful in computational research[9] since the musical elements (e.g. notes, durations, tempo, melodic structure, measure info) cannot be directly read by the machines.

In the scope of CompMusic, Karaosmanoğlu (2012) has created a music score collection called **SymbTr**. The scores in this collection are selected from reliable sources and further curated by experts. As it will be shown later in this Section, **SymbTr** collection is the biggest and most representative, machine-readable music score repository for OTMM. The collection is available online[10] and licensed under the Creative Commons Attribution-NonCommercial-ShareAlike 4.0 License (CC BY-NC-SA 4.0) (International).

The naming of the music scores follows a convention (referred as the **SymbTr**-slug throughout the text), which provides some of the key information to identify and categorize the scores. The structure is given as "[makam]--[form]--[usul]--[title]--[artist/location]," which basically provides the "slugs" of the makam, the form, the usul and the *title* (for vocal compositions). The last item (artist/region) is the *composer* for composed works, the *performer* for transcriptions, and the *transcriber* or the *region* for traditional folk pieces. For example, the **SymbTr**-slug of the composition *Ehl-i Aşkın Neşvegâhı*[11] is "kurdilihicazkar--sarki--agiraksak--ehl-i_as-

[9] An obvious exception is optical music recognition.

[10] https://github.com/MTG/SymbTr. Since v2.0.0, I am the principal maintainer of the repository and secondary curator after M. Kemal Karaosmanoğlu.

[11] http://musicbrainz.org/work/b43fd61e-522c-4af4-821d-db85722bf48c

kin--tatyos_efendi." Note that while almost all of the music scores in the **SymbTr** collection are related to compositions, there also exists performance transcriptions; e.g. "huseyni--oyunhavasi--nim-sofyan--cecen_kizi--tanburi_cemil_bey" is a transcription of Tanburi Cemil Bey's famous interpretation of the folk piece *Çeçen Kızı* (League, 2012).[12]

The first release of **SymbTr** collection is version 1.0.0.[13] This version is presented in (Karaosmanoğlu, 2012) and it consists of 1700 music scores from the folk and classical repertoires of OTMM. The second major version (v2.0.0) of the collection is released in June 2014[14] and described in (Karaosmanoğlu, 2015). In this version, there are 2200 music scores. As of October 2016, **SymbTr** v2.4.3 is the latest release.[15] The number of scores has not changed from v2.0.0 to v2.4.3, but there has been numerous improvements over the format consistency, content and metadata of the music scores. There are 155 unique makams, 56 unique forms, 88 unique usuls, 395 unique composer/transcribers (including Lâedrî and *traditional*) and 67 geographical regions annotated in the **SymbTr**-slugs.

Table 3.4 shows the top 10 makam, form, usul and composers in the **SymbTr** collection. The statistics in this Table generally overlap with the makam, form and usul statistics of the CompMusic OTMM audio collection shown in Table 3.2. In addition, our statistics coincide with the makam and usul statistics of TRT Tarihi Türk Müziği Arşivi) (English: TRT Historical Turkish Music Archive (**TRT-TTMA**) (accessed in 2005), which was reported by Çevikoğlu (2007).

The scores typically notate the basic melody of the composition devoid of the performance aspects such as intonation deviations and embellishments. The scores in the **SymbTr** collection are created by the software Mus2-alfa (Karaosmanoğlu, 2015, Appendix B)[16] and stored in several different formats (txt, mu2, MusicXML, MIDI and PDF). The formats are briefly explained below. For additional

[12] http://musicbrainz.org/recording/ed38dc73-7c0a-4362-84ca-07724ede9aab

[13] https://github.com/MTG/SymbTr/tree/v1.0.0

[14] https://github.com/MTG/SymbTr/releases/tag/v2.0.0

[15] https://github.com/MTG/SymbTr/releases/tag/v2.4.3

[16] not to be confused with the notation editor, Mus2

Makam	#	Form	#	Usul	#	Composer	#
Hicaz	157	Şarkı	994	Aksak	319	*Lâedrî/Traditional*	231
Nihavent	130	Türkü	285	Sofyan	293	Ahmet Avni Konuk	120
Uşşak	118	Seyir	169	Düyek	278	Şefik Gürmeriç	74
Rast	109	Küpe	120	Aksaksemai	128	Dede Efendi	71
Hüzzam	96	Peşrev	93	Curcuna	111	Erol Bingöl	66
Hüseyni	92	Sazsemaisi	86	Ağıraksak	108	Sadettin Kaynak	45
Segah	92	Aranağme	73	Semai	100	Hacı Arif Bey	40
Mahur	88	İlahi	42	Nimsofyan	99	Tanburi Cemil Bey	34
Hicazkar	79	Beste	41	Senginsemai	72	Rauf Yekta	32
Kürdilihicazkar	70	Yürüksemai	36	Türk aksağı	64	Şevki Bey	32
Other (145 makams)	1169	(46 forms)	261	(63 usuls)	628	(376 composers)	1260

Table 3.4: The most represented makams, forms, usuls and composers in the **SymbTr** collection and the corresponding number of instances.

Sira	Kod	Nota53	NotaAE	Koma53	KomaAE	Pay	Payda	Ms	LNS	Bas	Soz1	Offset
172	9	Si4b2	B4b1	312	313	2	8	500	95	96		24.2222222222
173	9	Sol4	G4	296	296	1	8	250	95	96	Sa	24.3333333333
174	9	Sol4	G4	296	296	1	8	250	95	96	kın*	24.4444444444
175	9	Sol4	G4	296	296	1	4	500	95	96	geç*	24.6666666667
176	9	Do5	C5	318	318	1	4	500	95	96	kal	24.8888888889
177	9	Do5	C5	318	318	1	8	250	95	96	ma*	25.0
178	9	Si4b2	B4b1	312	313	3	16	375	99	96	er	25.1666666667
179	9	La4	A4	305	305	1	16	125	95	96		25.2222222222
180	9	Sol4	G4	296	296	1	4	500	95	96	ken*	25.4444444444
181	9	Re5	D5	327	327	5	8	1250	100	96	gel**	26.0
182	9	Re5	D5	327	327	2	8	500	95	96		26.2222222222
183	9	Do5	C5	318	318	1	8	250	95	96	A	26.3333333333
184	9	Do5	C5	318	318	1	8	250	95	96	man*	26.4444444444
185	9	Si4b2	B4b1	312	313	1	8	250	99	96	geç*	26.5555555556
186	9	La4	A4	305	305	1	8	250	95	96		26.6666666667
187	9	La4	A4	305	305	1	8	250	99	96	kal	26.7777777778
188	9	Si4b2	B4b1	312	313	1	8	250	95	96		26.8888888889
189	9	Do5	C5	318	318	1	8	250	95	96	ma*	27.0
190	9	Si4b2	B4b1	312	313	3	16	375	99	96	er	27.1666666667
191	9	La4	A4	305	305	1	16	125	95	96		27.2222222222
192	9	Sol4	G4	296	296	1	8	250	99	96	ken*	27.3333333333
193	9	Si4b2	B4b1	312	313	1	16	125	99	96		27.3888888889
194	9	La4	A4	305	305	1	16	125	95	96		27.4444444444
195	9	La4	A4	305	305	5	8	1250	95	96	gel**	28.0
196	9	Do5	C5	318	318	1	12	167	95	96	ARANAĞME	28.0740740741
197	9	Si4b2	B4b1	312	313	1	12	167	95	96	.	28.1481481481
198	9	La4	A4	305	305	1	12	167	95	96	.	28.2222222222
199	9	Sol4	G4	296	296	1	12	167	95	96	.	28.2962962963
200	9	Fa4#3	F4#4	290	291	1	12	167	95	96	.	28.3703703704
201	9	Fa4#3	F4#4	290	291	1	12	167	95	96	.	28.3703703704

Table 3.5: The contents of the **SymbTr**-txt file of the composition *Bu Akşam Gün Batarken Gel* from the 172th note to the 201th note. The spaces are displayed as "*" for visualization purposes.

information about the **SymbTr** collection and the score formats, please refer to (Karaosmanoğlu, 2012) and (Karaosmanoğlu, 2015).

- The data in the **SymbTr**-txt scores are stored as "tab separated values," where each row is a note or an editorial annotation (such as usul change) and each column represents an attribute such as the note symbol, the duration, the measure

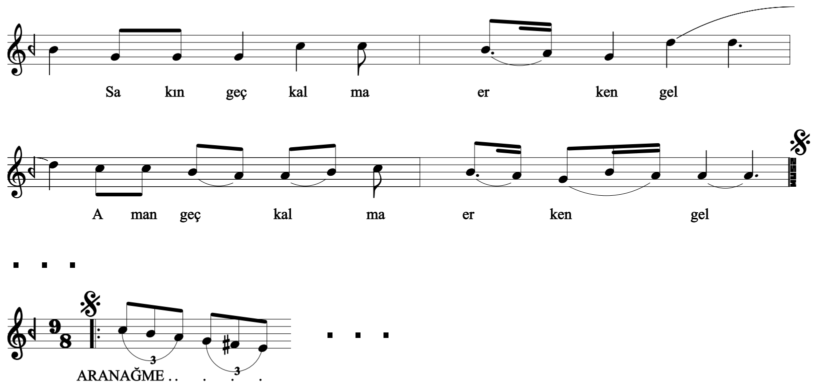
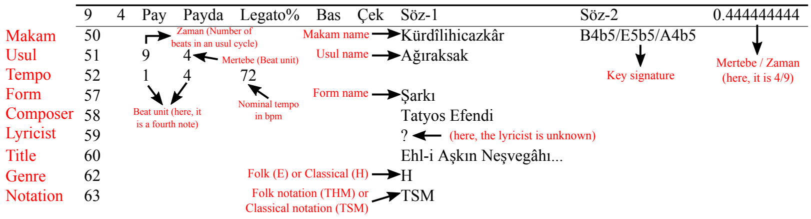


Figure 3.2: The visualization of the note sequence in Table 3.5 on the **SymbTr**-PDF file.



	9	4	Pay	Payda	Legato%	Bas	Çek	Söz-1	Söz-2	0.444444444
Makam	50							Kürdîlihicazkâr	B4b5/E5b5/A4b5	
Usul	51		9	4				Ağıraksak		
Tempo	52		1	4	72					
Form	57							Şarkı		
Composer	58							Tatyos Efendi		
Lyricist	59							?		
Title	60							Ehl-i Aşkın Neşvegâhı...		
Genre	62							H		
Notation	63							TSM		

Zaman (Number of beats in an usul cycle)
Mertebe (Beat unit)
Beat unit (here, it is a fourth note)
Nominal tempo in bpm
Makam name
Usul name
Form name
(here, the lyricist is unknown)
Folk (E) or Classical (H)
Folk notation (THM) or Classical notation (TSM)
Key signature
Mertebe / Zaman (here, it is 4/9)

Figure 3.3: The metadata header of the **SymbTr**-mu2 score *Ehl-i Aşkın Neşvegâhı*. The annotations are shown in red.

marking or the lyrics. The rests (rows with the value “Es” in the note columns), usul alterations (rows with the value 51 in the *Kod* column, indicating an usul change within the piece) and measure marking columns (the *Offset* column) are introduced in **SymbTr** v2.0.0.

Table 3.5 shows a short fragment selected from the **SymbTr**-txt score of the composition *Bu Akşam Gün Batarken Gel*.[17] The note symbols and pitch intervals are given according to the 24-TET system defined in the AEU theory in the *NotaAE* column or the 53-TET system in the *Nota53* column.[18] The note symbols are encoded as “[note-symbol][octave](accidental)(comma),” i.e. a “B4b1” note in the *NotaAE* column

[17] http://musicbrainz.org/work/30cdf1c2-8dc3-4612-9513-a5d7f523a889

[18] The unit interval of the 53-TET, which is simply the 1/53th of an octave, is called a Holderian comma (Hc).

refers to the segah note according to AEU theory (Table 3.5). The lyrics are synchronous to the note onsets on the syllable level. The final syllable of each word ends with a single space and the final syllable of each poetic line ends with double spaces. Some columns may be overloaded with additional types of information. For example, the lyrics (the *Soz1* column) column also contains editorial annotations entered in capital letters such as the section names, instrumentation and tempo changes (e.g. the lyrics of the 196th note in Table 3.5 marks the start of the Aranağme section).

The default score output of Mus2-alfa is the **SymbTr**-txt format. In the automatic analysis (Chapters 4 and 6), the scores in the txt format are used, as they are the reference format.

- The PDF files in the **SymbTr** repository reflect the same sequence of events in the **SymbTr**-txt scores. Figure 3.2 shows the PDF representation of the same sequence in Table 3.5 selected from the composition *Bu Akşam Gün Batarken Gel*.

- Likewise, the MIDI-scores follow from the **SymbTr**-txt files. They retain the tuning information in the form of pitch bends.

- Introduced in the **SymbTr** release version 2.0.0, mu2 is a file format that can be read by Mus2.[19] Mus2 is a music notation software for OTMM, which supports makam music concepts and microtonal playback. The note sequences in the **SymbTr**-txt and **SymbTr**-mu2 scores are identical. The data organization in the mu2 format is very similar to the one in the **SymbTr**-txt, with several improvements, e.g. on rhythmic and lyrics organization. In addition, the mu2-scores start with a easy-to-parse metadata header. Figure 3.3 shows the header of the mu2 file of the composition *Ehl-i Aşkın Neşvegâhı*.[20]

[19] https://www.mus2.com.tr/en/

[20] https://github.com/MTG/SymbTr/blob/v2.4.3/mu2/kurdilihicazkar--sarki--agiraksak--ehl-i_askin--tatyos_efendi.mu2

- The MusicXML v3.0 files are introduced in **SymbTr** v2.0.0. MusicXML[21] is a format which can be imported and exported by well-known music notation software such as MuseScore, Finale and Sibelius. The **SymbTr**-MusicXML files are generated from the music sequence in the **SymbTr**-txt files and the editorial metadata in the **SymbTr**-mu2 files. The conversion process is explained in detail in Section 4.4.2.

In addition, we have built tools to convert the MusicXML scores to LilyPond and scalable vector graphics format (SVG) formats. The conversion is explained in Section 4.4.2. Independently, the txt files in the **SymbTr** releases are converted to *abc notation*[22] by Seymour Shilen.[23] These formats are not distributed in the official **SymbTr** collection.

Now we present the coverage, completeness and quality and reusability of the **SymbTr** collection as discussed in (Uyar et al., 2014).

Coverage

To the best of our knowledge there are only two machine-readable score collections of OTMM apart from **SymbTr** collection, which can be used for computational research. First is the Uzun Hava Humdrum Database (**UHHD**) (Şentürk, 2011). This collection features the 77 music scores of *uzun hava*s, a non-metered improvisational form of Turkish folk music. Due to its specialized nature, this collection is not considered for comparison. A more relevant collection is the Türk Sanat Müziği Derlemi (English: Turkish Art Music Corpus) (**TSMD**) (Atalay & Yöre, 2011), which includes 600 compositions equally divided into 20 makams (i.e. 30 pieces per makam). It is smaller than the **SymbTr** collection.

To get a better means of comparison, we also refer to the online music score collections, in which the music scores are stored in various image formats. Although these collections are not machine-readable, hence unsuitable for computational research, they contain a much greater amount of music scores with respect to the

[21] http://www.musicxml.com/

[22] http://abcnotation.com/

[23] http://ifdo.ca/~seymour/runabc/makams/index.html

machine-readable collections. This leads us to accept these collections as our references while measuring the coverage of the score collection. Among these online collections, we selected TRT Tarihi Türk Müziği Arşivi) (English: TRT Historical Turkish Music Archive (**TRT-TTMA**)[24] and the Türk Müzik Kültürünün Hafızası (English: "Memory of Turkish Music Culture" Collection) (**TMKH**) collections.[25] From these collections, we crawled the metadata of the music scores to obtain the statistics (Table 3.6).

TRT-TTMA is arguably the most reliable resource as it originates from TRT. The scores in **TRT-TTMA** are sold online. As of July 2014, **TRT-TTMA** includes $\sim$ 17000 scores in total, all of which are manually scanned from physical scores. **TRT-TTMA** has some duplicate entries and some compositions, which are not in the context of OTMM, *(e.g. church chants, operettas etc.)*. When these compositions are removed from comparison, the number of compositions are reduced to $\sim$ 12000.

TMKH is created by funds from the *Istanbul 2010 European Capital of Culture Organization*[26] through the *European Union*. As of July 2014, the **TMKH** collection includes $\sim$ 45000 scanned scores (where multiple versions are available for almost each work) of the personal collections of 3 professional makam musicians/scholars. The collection is free, however there are several restrictions on the site navigation and the number of daily downloads.

Some of the names of the makam, usul and form in the collections slightly differ from each other. To match the names, we carry a semi-automatic procedure. First, we use an automatic string matching method. The algorithm we chose uses a weighted measure which consists of two edit distance measures:[27] longest common subsequence, and Damerau–Levenshtein distance, which have 0.7 and 0.3 weights respectively. The weights are determined empirically by varying them to find a configuration that results in satisfactory matches. Then, we go through the meatches and manually correct the erroneous and missing pairs.

[24] http://www.trtkulliyat.com/

[25] http://www.sanatmuziginotalari.com/; accesible through http://turkmusikisivakfi.org/.

[26] http://istanbul2010.org/

[27] The implementation is here: https://github.com/gopalkoduri/string-matching/

To assess how well the **SymbTr** collection covers the OTMM, we compared the version v2.0.0 of our collection against these music score collections. From each collection we report the number of compositions, composers, makams, forms and usuls. We also check how much the makams, forms and usuls in the **SymbTr** collection overlap with the corresponding type of metadata in other collections. We define *overlap* as:

$$\mathscr{O}\left(\mathscr{A}^{(\mathcal{R})}\right) = \frac{\left|\mathscr{A}^{(\mathbf{SymbTr})} \cap \mathscr{A}^{(\mathcal{R})}\right|}{\left|\mathscr{A}^{(\mathcal{R})}\right|} \tag{3.1}$$

where $\mathscr{A}^{(\mathbf{SymbTr})}$ is the set of the subjected attribute (makam, usul or form) from the score collection **SymbTr** v2.0.0, $\mathscr{A}^{(\mathcal{R})}$ is the set of the subjected attribute from the referance collection $\mathcal{R}$, against which we want to measure our collection's coverage and $\mathscr{O}\left(\mathscr{A}^{(\mathcal{R})}\right)$ is the overlap, which demonstrates how much the subjected attribute of $\mathcal{R}$ is represented in **SymbTr**.

Table 3.6 shows the overlap of the makams, usuls, forms between our collection and the three music score collections explained above. We can observe that the **SymbTr** collection covers almost all of the makams, usuls and forms in the **TSMD**. While the number of compositions are much less than **TRT-TTMA** and **TMKH**, there is a fair number of overlapping makams, usuls and forms the **SymbTr** collection with respect to **TRT-TTMA** and **TMKH**. Note that in OTMM, it is common to have different titles for the scores of the same composition (first line of lyrics, the chorus etc.) and a composer might have various names (e.g. aliases, titles, added surname etc.). It is hard to obtain an accurate overlap for these attributes. Hence, the overlap of the composers and the compositions are not computed.

Note that the makams, usuls and forms listed in the score collections are not evenly distributed, some of these attributes are much more represented than the others. Hence we should also consider the coverage of these attributes with respect to their rate of presence in the reference collection. Taking these circumstances into consideration, we have modified the overlap function by adding some measure parameters as explained below:

	SymbTr v2.0.0	**TSMD**	**TRT-TTMA**	**TMKH**
Compositions	2,200	600	12,035	45,368
Composers	455	230	1,447	2,674
Makams	157	20 (1)	293 (0.49)	317 (0.45)
Usuls	84	46 (0.89)	N/A	382 (0.22)
Forms	62	6 (1)	110 (0.35)	90 (0.31)

Table 3.6: Coverage of the score collection in the corpus. The number in paranthesis is the overlap measure Equation 3.1 in percentage. N/A indicates that data is not available.

- The attribute set $\mathscr{A}^{(\boldsymbol{\mathcal{R}})}$ is treated as an enumeration such that each element $\alpha_k^{(\boldsymbol{\mathcal{R}})} \in \mathscr{A}^{(\boldsymbol{\mathcal{R}})}$ is ordered according its occurance (in decreasing order) with respect to the other elements. The number of elements in $\mathscr{A}^{(\boldsymbol{\mathcal{R}})}$ is $\left|\mathscr{A}^{(\boldsymbol{\mathcal{R}})}\right|$.

- An element $\alpha_k^{(\boldsymbol{\mathcal{R}})}$ has an occurrence, $o\left(\alpha_k^{(\boldsymbol{\mathcal{R}})}\right)$, in the collection such that $o\left(\alpha_k^{(\boldsymbol{\mathcal{R}})}\right) \geq o(\alpha_{k+1}^{(\boldsymbol{\mathcal{R}})}), \forall k \in \left[1 : \left|\mathscr{A}^{(\boldsymbol{\mathcal{R}})}\right| - 1\right]$. Moreover,

$$\sum_{k=1}^{\left|\mathscr{A}^{(\boldsymbol{\mathcal{R}})}\right|} o\left(\alpha_k^{(\boldsymbol{\mathcal{R}})}\right) = |\boldsymbol{\mathcal{R}}| \tag{3.2}$$

  where $|\boldsymbol{\mathcal{R}}|$ indicates the number of scores in the reference collection, $\boldsymbol{\mathcal{R}}$.

- $\mathscr{A}^{(\boldsymbol{\mathcal{R}})}[1 : k]$ denotes the first k elements in the enumerated attribute set, i.e. the k most occuring elements in the attribute set.

- The occurence ratio, $\hat{o}\left(\alpha_k^{(\boldsymbol{\mathcal{R}})}\right)$, is defined as:

$$\hat{o}\left(\alpha_k^{(\boldsymbol{\mathcal{R}})}\right) = \frac{o\left(\alpha_k^{(\boldsymbol{\mathcal{R}})}\right)}{|\boldsymbol{\mathcal{R}}|} \tag{3.3}$$

- The cumulative occurrence ratio $\hat{O}\left(\alpha_k^{(\boldsymbol{\mathcal{R}})}\right)$ is the summation of the occurence ratios of the elements in the enumeration $\mathscr{A}^{(\boldsymbol{\mathcal{R}})}[1:k]$ up to $\alpha_k^{(\boldsymbol{\mathcal{R}})}$ as,

$$\hat{O}\left(\alpha_k^{(\boldsymbol{\mathcal{R}})}\right) = \frac{\sum_1^k o\left(\alpha_k^{(\boldsymbol{\mathcal{R}})}\right)}{|\boldsymbol{\mathcal{R}}|} \tag{3.4}$$

Notice that, $\hat{O}\left(\alpha_{|\mathscr{A}^{(\boldsymbol{\mathcal{R}})}|}^{(\boldsymbol{\mathcal{R}})}\right) = 1$, as it includes all of the reference collection.

- We measure the overlap of **SymbTr** against $\hat{O}\left(\alpha_k^{(\boldsymbol{\mathcal{R}})}\right)$'s.

$$\mathscr{O}_k\left(\mathscr{A}^{(\boldsymbol{\mathcal{R}})}\right) = \frac{\left|\mathscr{A}^{(\textbf{SymbTr})} \cap \mathscr{A}^{(\boldsymbol{\mathcal{R}})}[1:k]\right|}{|\mathscr{A}^{(\boldsymbol{\mathcal{R}})}[1:k]|} \tag{3.5}$$

- Finally we define the attribute coverage $\mathscr{C}\left(\mathscr{A}^{(R)}\right)$ of **SymbTr** against $\boldsymbol{\mathcal{R}}$.

$$\mathscr{C}(\mathscr{A}^{(R)}) = max\left(\hat{O}(\alpha_k^{(\boldsymbol{\mathcal{R}})})\right) \quad | \quad \mathscr{O}_k\left(\mathscr{A}^{(\boldsymbol{\mathcal{R}})}\right) = 1 \tag{3.6}$$

By applying this modified procedure, we have reached detailed results specifically different for the entities with different occurence ratios. In Figure 3.4, two functions for $\mathscr{O}_k\left(makams^{(\boldsymbol{\mathcal{R}})}\right)$ are provided with respect to $\mathscr{A} = makams$. In the Figure, **SymbTr** corresponds to v2.0.0 of our collection and $\boldsymbol{\mathcal{R}}$ corresponds to either **TMKH** or **TRT-TTMA**.

The overlap, $\mathscr{O}_k\left(makams^{(\textbf{TMKH})}\right)$, of **TMKH** by the **SymbTr** collection is equal to 1, when the makams which contribute less then 0.1% to the **TMKH** are excluded, i.e. $\hat{o}\left(makam_k^{(\textbf{TMKH})}\right) < 0.001$. According to this specific k value, **SymbTr** covers **TMKH** by 96%; $\mathscr{C}\left(makams^{(\textbf{TMKH})}\right) = 0.96$.

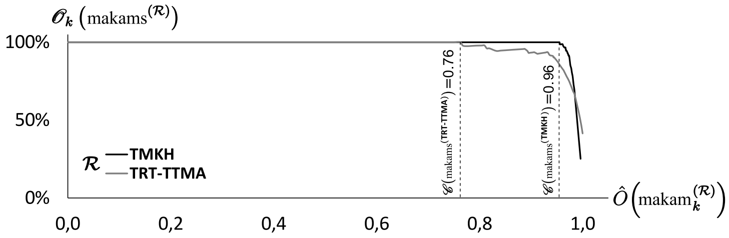


Figure 3.4: Overlap with respect to the makam, our corpus vs **TMKH** and **TRT-TTMA**. The dashed lines indicate the coverage values for **TRT-TTMA** (0.76) and **TMKH** (0.96).

The overlap, $\mathscr{O}_k\left(makams^{(\textbf{TRT-TTMA})}\right)$ of **TRT-TTMA** by the **SymbTr** collection is equal to 1, when the makams which contribute less than 0.6%, i.e. $\hat{o}\left(makam_k^{(\textbf{TRT-TTMA})}\right) < 0.006$, are excluded, providing a coverage value $\mathscr{C}\left(makams^{(\textbf{TRT-TTMA})}\right)$ of $= 0.76$.

For the form attribute, the coverage $\mathscr{C}\left(forms^{(\textbf{TMKH})}\right)$ is 98% with $\hat{o}\left(form_k^{(\textbf{TMKH})}\right) < 0.002$ are excluded for **TMKH**. For **TRT-TTMA**, the form coverage $\mathscr{C}\left(forms^{(\textbf{TRT-TTMA})}\right)$ is 86%, when the forms with $\hat{o}\left(form_k^{(\textbf{TRT-TTMA})}\right) < 0.01$ are excluded. For **TMKH**, **SymbTr** has a usul coverage $\mathscr{C}\left(usuls^{(\textbf{TMKH})}\right)$ of 94%, when $\hat{o}\left(usul_k^{(\textbf{TMKH})}\right) < 0.003$ are excluded. **TRT-TTMA** does not provide the usul information.

Completeness

The **SymbTr**-slugs include the makam, usul, form and *artist* information. Moreover, miscellaneous metadata such as the key signature and nominal tempo can be easily parsed from mu2 headers or crawled from MusicBrainz by referring to the relevant work/recording MusicBrainz identifier (MBID).[28]

On the other hand, the set of structure annotations in the lyrics column in the **SymbTr**-txt scores (and all other **SymbTr** formats, which are generated by Mus2-alfa) does not convey the complete

[28] The relations are stored in: `https://github.com/MTG/SymbTr/blob/v2.4.3/symbTr_mbid.json`.

information about the section boundaries and the section names. First, the section name (and hence the first note of a section) is only given for the instrumental sections and the final note of the instrumental sections are not marked at all. Moreover, the section name does not indicate if there are any differences between the renditions of the same section. For the vocal sections, only the last syllable of a poetic line is marked (i.e. with double space in the end of the lyrics syllable). The marked note does not typically coincide with the actual ending of the vocal section since a syllable can be sung for longer than one note or there might be a short instrumental movement in the end of the vocal section.

Out of 2200, 1771 txt-scores in **SymbTr** v2.4.3 has some editorial section information. The remaining 429 scores either lack the editorial section information or they are very short compositions such that they do not have any sections. Given the linked and/or embedded metadata, we can argue that **SymbTr**-scores are editorially complete except the section labels. Later in Section 4.3 an automatic section extraction and labeling method is proposed, which uses the implicit section information in the **SymbTr**-txt scores.

Including the transcription of Tanburi Cemil Bey's *Çeçen Kızı* performance, 856 **SymbTr** scores are currently paired with 2217 recordings in the CompMusic OTMM audio collection. The relational statistics between the audio recordings and **SymbTr** scores can be seen in Table 3.7.

Quality and Re-usability

The scores in the **SymbTr** collection are obtained from reliable sources and curated by experts (Karaosmanoğlu, 2012; Karaosmanoğlu, 2015). Moreover, all music scores in the **SymbTr** collection contain the OTMM-specific information observed in the notation properly such as the key signature and accidental symbols so that the (machine-readable) scores can be sythesized with proper tuning.

The **SymbTr** collection is available to public and licensed under the CC BY-NC-SA 4.0 (International). This way, a user can make necessary changes on a certain score or contribute his/her own scores and share their own works under the same license.

n	num. works with **SymbTr**, related to *n* num. recordings	total num. audio recordings
1	315	315
2	191	382
3	130	390
4	84	336
5	41	205
6	29	174
7	24	168
8	9	72
9	7	63
10	3	30
11	4	44
12	2	24
13	1	13
Total	855	2216

Table 3.7: Number of works with **SymbTr**-scores distributed with respect to the related number of audio recordings.

The MusicXML is supported by many popular notation software. There are also numerous score parsing and analysis tools,[29] which can read and write this format. The MusicXML format is highly suitable for music score creation, sharing and content validation. We plan to make MusicXML as the default score format of **SymbTr** in the future (v3.0.0).Metadata

3.1.3 Music Theory

The makam, form and usul information is fetched from **SymbTr**-slugs, **SymbTr**-txt files, **SymbTr**-mu2 headers, MusicBrainz and the internal music theory library of Mus2-alfa (Karaosmanoğlu, 2015, Appendix B) (courtesy of M. Kemal Karaosmanoğlu). The makam, form and usul instances are matched with each other by using the semi-automatic string matching procedure explained in the

[29] e.g. in Music21: http://web.mit.edu/music21/doc/usersGuide/usersGuide_08_installingMusicXML.html

score coverage computation (Section 3.1.2). The makam,[30] form[31] and usul[32] dictionaries are stored separately as JSON files. Each instance in the resultant dictionaries contains the naming of the instance in different sources and the structured music theory knowledge obtained from the music theory library of Mus2-alfa. An instance with a unique identifier is created in Dunya-makam for each makam form and usul instance. The instances also consist of aliases (e.g. *Çargah (Yeni)* vs. *Yeni Çargah*).

Figure 3.5 shows an example instance from each dictionary and the explanation of the structured data. The dictionary is used later in automatic score validation (Section 4.4.3) and also to supply the basic music theory information in automatic description methods (e.g. Section 5.9 and Section 6.4).

3.1.4 Metadata

The metadata includes the general information about our corpus. It identifies the entities forming the corpus such as the audio recordings, releases, artists (composers, lyricists, performers etc.), makams, usuls, forms and works. Moreover, these entities are *linked* with each other (e.g. the *instrument* an *artist* performs in an *audio recording*) and also with other information sources such as related web pages (e.g. artist biographies) and knowledge bases (e.g. Wikidata[33]). These relationships may be used to navigate and discover the concepts of OTMM (e.g. in Dunya-makam as will be described in Section 7.1.1).

As of February 2016, we have collected more than 15000 instances of metadata and 92000 relationships (Figure 3.6). These include the recording, release, artist, work, instrument, makam, form and usul information in the relevant entities in our corpus. Some

[30] `https://github.com/sertansenturk/otmm_corpus_stats/blob/94c05d4e08486012f43b268955f3f1b51a3658fb/data/makamFormUsulDicts/makam_extended.json`

[31] `https://github.com/sertansenturk/otmm_corpus_stats/blob/94c05d4e08486012f43b268955f3f1b51a3658fb/data/makamFormUsulDicts/form_extended.json`

[32] `https://github.com/sertansenturk/otmm_corpus_stats/blob/94c05d4e08486012f43b268955f3f1b51a3658fb/data/makamFormUsulDicts/usul_extended.json`

[33] `https://www.wikidata.org`

```
                                   Description                 JSON Representation
                                      makam key | "huzzam": {
                             name in Dunya-makam |     "dunya_name": "Hüzzam",
                  unique identifier in Dunya-makam |     "dunya_uuid": "c5fa8f01-6959-4e6d-a998-d31d0fc17182",
                 tonic frequency when A4 = 440 Hz |     "karar_midi_freq": 487.46,
                                   tonic symbol |     "karar_symbol": "B4b1",
                 accidentals in the key signature |     "key_signature": [
                                                |         "B4b1",
Makam                                           |         "E5b4",
                                                |         "F5#4"
                                                |     ],
                 tags in MusicBrainz recordings |     "mb_tag": [
                                                |         "hüzzam"
                                                |     ],
                       name in SymbTr-mu2 files |     "mu2_name": "Hüzzam",
                           name in SymbTr-slug |     "symbtr_slug": "huzzam"
                                                |   }
-------------------------------------------------------------------------------------------
                                       form key | "sazsemaisi": {
                             name in Dunya-makam |     "dunya_name": "Sazsemaisi",
                  unique identifier in Dunya-makam |     "dunya_uuid": "9a33c7d6-7fe4-485d-aea6-0af3673e9ac1",
                       name in SymbTr-mu2 files |     "mu2_name": "Sazsemâîsi",
     repetitive section name(s) in SymbTr-txt files |     "repetitive_section": "TESLIM,TESLİM,MÜLÂZİME,MULAZIME",
Form                       name in SymbTr-slug |     "symbtr_slug": "sazsemaisi",
                 tags in MusicBrainz recordings |     "mb_tag": [
                                                |         "sazsemaisi"
                                                |     ],
                 "vocal" or "instrumental" type |     "type": "instrumental",
      genre ("TSM" for classical, "THM" for folk) |     "genre": "TSM"
                                                |   }
-------------------------------------------------------------------------------------------
                                       usul key | "semai": {
                             name in Dunya-makam |     "dunya_name": "Semai",
                  unique identifier in Dunya-makam |     "dunya_uuid": "1942822b-778c-46a0-ad68-a38fdf92e321",
                     list of variants of the usul |     "variants": [
                                                |         {
        duration of the strokes  in the usul cycle |             "clustering": [
                                                |                 1,
                                                |                 1,
                                                |                 1
                                                |             ],
                 ID of the variant in Mus2-alfa |             "symbtr_internal_id": "4",
        name of the variant in SymbTr-mu2 files |             "mu2_name": "Semâî",
               slug of the variant in Mus2-alfa |             "symbtr_internal": "semai",
                      mertebe of the variant |             "mertebe": 4.0,
         zaman (number of pulses) of the variant |             "num_pulses": 3.0
                                                |         },
                                                |         {
Usul                                            |             "clustering": [
                                                |                 2,
                                                |                 2,
                                                |                 2
                                                |             ],
                                                |             "symbtr_internal_id": "97",
                                                |             "mu2_name": "Semâî (3/8)",
                                                |             "symbtr_internal": "semai_3_8",
                                                |             "mertebe": 8.0,
                                                |             "num_pulses": 3.0
                                                |         }
                                                |     ],
                           name in SymbTr-slug |     "symbtr_slug": "semai",
                 tags in MusicBrainz recordings |     "mb_tag": [
                                                |         "semai"
                                                |     ]
                                                |   }
```

Figure 3.5: Examples of makam, form and usul instances in the dictionaries stored in JSON format and their description.

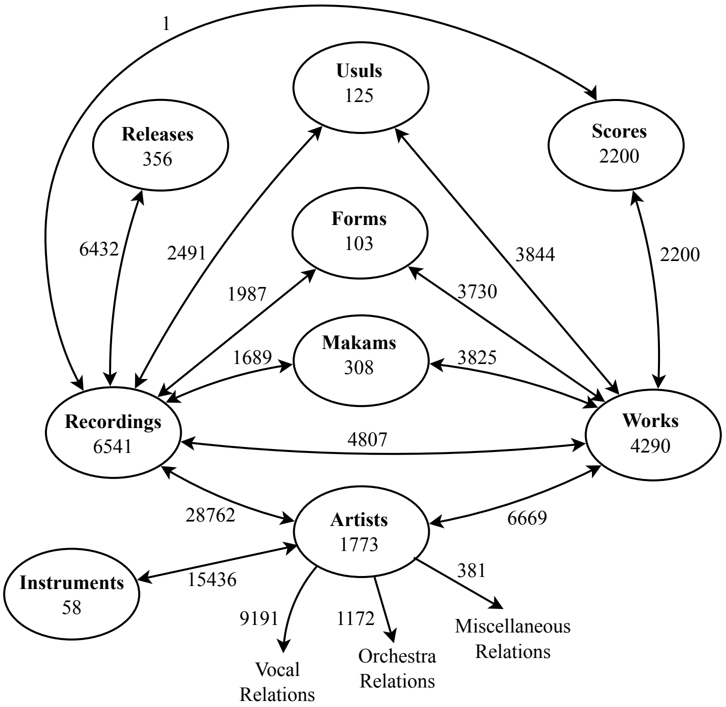


Figure 3.6: The number of metadata instances and the number of relationships between each type of entity in the CompMusic OTMM makam corpus.

entries in MusicBrainz also include textual annotations within the structural data.[34]

3.1.5 Automatic Description

As explained in earlier in Section 3.1, the automatic description is considered as part of the corpora. As of any content in the corpus, the automatic description is not static since the analysis tools (and hence results) can be improved, extended and modified in the future. Currently, the total size of the automatic description is approximately 191 gigabytes.

[34] e.g. http://musicbrainz.org/work/93f31506-25aa-49da-96cd-660c6a2e44bf and http://musicbrainz.org/recording/37dd6a6a-4c19-4a86-886a-882840d59518

The automatic description methodologies and the analysis of the CompMusic OTMM corpus are described between the Chapters 4 and 6. Please refer to these Chapters for more detail and Chapter 7 for how the automatic description is used to discover the CompMusic OTMM corpus.

3.2 Test Datasets

Test datasets are collections arranged for the specific research problems. These datasets are typically used as the ground-truth to evaluate methodologies applied to certain problems. They can be composed of different types of data such as synthetic or "real-world" data with manual, semi-automatic or automatic annotations.

In our test datasets, we have manual annotations by the experts. The data is selected from the CompMusic OTMM corpus. Bozkurt, Ayangil, and Holzapfel (2014) made a review of computational analysis literature for OTMM. The datasets that we mention in this section are useful for some of the research tasks discussed in this paper such as structure analysis, automatic tonic identification, automatic ornamentation segmentation and melodic phrase segmentation. All the test datasets mentioned in this Section are either the first datasets or the first open datasets created for the studied research problems on OTMM.

3.2.1 Symbolic Melodic Segmentation Dataset

Karaosmanoğlu and Bozkurt have studied the problem of usul and makam driven automatic melodic segmentation for Turkish Music in (Bozkurt, Karaosmanoğlu, Karaçalı, & Ünal, 2014). The source code and the test dataset are published in (Karaosmanoğlu, Bozkurt, Holzapfel, & Doğrusöz Dişiaçık, 2014) and they are available online.[35] The test dataset presents a large machine-readable dataset of OTMM scores in **SymbTr** v2.0.0 format, segmented into phrases. The segmentation facilitates computational research on melodic similarity between phrases, and relation between melodic phrasing and meter, rarely studied topics due to unavailability of data resources.

[35] http://www.rhythmos.org/shareddata/turkishphrases.html

The phrase segmentation code and the test dataset are used as part of the automatic score structural analysis methodology described in Section 4.3. To simplify the integration process, I forked the dataset to GitHub.[36] The latest release has minor changes from the original test dataset in (Karaosmanoğlu et al., 2014) such as duplicate file removal and changing the encoding of the scores to UTF-8. The dataset currently consists of 31362 phrases on a set of 480 scores of different compositions manually annotated by 3 experts of this music.

3.2.2 Symbolic Section Dataset

To test the semiotic labeling method in described in (Şentürk & Serra, 2016b) (Section 4.3), I have created a small dataset by marking the start and end of each section in the selected **SymbTr**-scores, and labeling the melodic and lyric relations manually as described in Section 4.3.3. The dataset is open and available online.[37]

The release published for (Şentürk & Serra, 2016b) contains the **SymbTr** scores of 23 vocal compositions in the şarkı form and 42 instrumental compositions in peşrev and sazsemaisi forms in the txt and PDF formats. The scores are selected from the **SymbTr** release version 2.4.2.

3.2.3 Makam Recognition Dataset

We have created a comprehensive dataset to address the lack of open and representative datasets for makam recognition.[38] The release published in (Karakurt, Şentürk, & Serra, 2016) is composed of 50 recordings from each of the 20 most common makams in CompMusic OTMM corpus. Currently, this release is the largest makam recognition dataset. The dataset provides the manually annotated tonic and makam, and also the predominant melody, which is used in the makam recogition and tonic identification experiments conducted in (Karakurt et al., 2016) (Section 5.7.3). The dataset is explained in detail in Section 5.7.4.

[36] https://github.com/MTG/otmm_symbolic_phrase_dataset

[37] https://github.com/MTG/otmm_symbolic_section_dataset/releases/tag/fma_2016

[38] https://github.com/MTG/otmm_makam_recognition_dataset

3.2.4 Tonic Identification Datasets

The tonic identification task has been studied in (Şentürk, Gulati, & Serra, 2013) (Section 6.4), (Atlı et al., 2015) (Section 5.7.2) and (Karakurt et al., 2016) (Section 5.7.2). Additionally, the score-informed composition identification method proposed in (Şentürk & Serra, 2016a) (Section 6.6) identifies the tonic in the process.

For score-informed tonic identification (Şentürk et al., 2013), the tonic frequency of 257 audio recordings are annotated (Section 6.4.3).[39] The **SymbTr**-score (version $v1.0.0$) of the 57 relevant compositions are indicated in the dataset. These recordings and the scores are identical to the ones in the section linking dataset explained in Section 3.2.6. Later in (Atlı et al., 2014), the tonic of 1093 audio recordings are annotated.[40] The test datasets used in (Karakurt et al., 2016) and (Şentürk & Serra, 2016a) are introduced in Section 3.2.3 and Section 3.2.5, respectively.

Recently, the tonic annotations included in all these test datasets are combined together.[41] The combined dataset consists of tonic annotations of 2007 recordings. Instead of a single annotation per recording, the combined dataset stores all the tonic annotations for a recording with its source. While approximately three fourth of the dataset is only annotated in a single source so far; we plan to apply the automatic tonic identification methods described in (Şentürk et al., 2013; Atlı et al., 2015) and (Karakurt et al., 2016) to the recordings in the test dataset. Similar to (Holzapfel, Davies, Zapata, Oliveira, & Gouyon, 2012), we would like to use the mutual (dis)agreement between the manual and automatic annotations for each recording not only to evaluate the annotations themselves but also to label "difficult" performances and investigate possible causes (makam, heterophony, production quality etc.).

[39] `https://github.com/MTG/otmm_tonic_dataset/releases/tag/2013_ismir`

[40] `https://github.com/MTG/otmm_tonic_dataset/releases/tag/2015_fma`

[41] `https://github.com/MTG/otmm_tonic_dataset/releases/tag/senturk2016thesis`

3.2.5 Composition Identification Dataset

For the score-informed composition identification experiments in (Şentürk & Serra, 2016a) (explained in Section 6.6), a test dataset of the music scores of 147 instrumental compositions selected from the **SymbTr** collection and 743 audio recordings selected from the CompMusic OTMM audio collection are collected.[42] In the dataset, there are 360 recordings associated with 87 music scores, forming 362 relevant audio-score pairs.

The predominant melody is included in the dataset for each recording. Moreover, the tonic frequency of each audio score pair is annotated manually. The dataset and the experiments are explained more in detail in Section 6.6.3 and Section 6.6.4, respectively.

3.2.6 Section Linking Dataset

To test the section linking methodology proposed in (Şentürk, Holzapfel, & Serra, 2014) (explained in Section 6.7), we have annotated the start and end of each section in 57 **SymbTr**-scores and 257 relevant audio recordings.[43] The number of section annotations in the audio recordings is 2095. The dataset is explained more in detail in Section 6.7.3. The dataset is also used partially in (Holzapfel et al., 2015) (Section 6.7.6).

3.2.7 Partial Audio-Score Alignment Dataset

We have recently created a dataset to study the time-deviations between the human annotators and automatic audio-score alignment algorithms.[44] The dataset consists of short fragments (< 1 minute) selected from 19 audio recordings in the CompMusic OTMM corpus. The recordings are associated with 14 **SymbTr**-scores. Each recording excerpt is annotated at least by 4 experts by referring to the note sequence in the relevant **SymbTr**-score. Note that the an-

[42] https://github.com/MTG/otmm_composition_identification_dataset/tree/smc2016

[43] https://github.com/MTG/otmm_section_dataset/tree/2014_jnmr

[44] https://github.com/MTG/otmm_partial_alignment_dataset

notations (of 1 expert) are partially taken from (Benetos & Holzapfel, 2015).

We are currently preparing the experimental setup to study the aforementioned problem. In the meantime, Atlı (2016) used a part of the dataset to the evaluate of the predominant melody extraction method proposed in (Atlı et al., 2014) (Section 5.2).

3.2.8 Audio-Score Alignment Dataset

For the initial experiments in note-level audio-score alignment (Section 6.8), we collected 6 audio recordings of 4 peşrev compositions (Şentürk, Gulati, & Serra, 2014). The audio recordings in the dataset have the annotated tonic frequencies, 51 section annotations and 3896 note annotations in total. The note annotations are derived from the manual transcriptions done in (Benetos & Holzapfel, 2015) and they follow the note sequences in the corresponding **SymbTr**-scores. The statistics of this dataset is given in Section 6.8.2.[45]

Currently, we are extending the dataset by including the rest of the transcriptions in (Benetos & Holzapfel, 2015) and also annotating additional recordings.

3.2.9 Audio-Lyrics Aligment Dataset

Within the CompMusic project, Dzhambazov et al. has been working on automatic lyrics-to-audio alignment in OTMM (Dzhambazov, Şentürk, & Serra, 2014; Dzhambazov & Serra, 2015; Dzhambazov, Srinivasamurthy, Şentürk, & Serra, 2016). The *Acapella Sections Dataset*[46] and *Şarkı Vocal Dataset*,[47] used in these studies utilize the manually annotated sections in the section linking dataset (Section 3.2.6). *Şarkı Vocal Dataset* (Dzhambazov et al., 2016) also incorporates the note-level annotations in the latest version of the OTMM audio-score alignment dataset (Section 3.2.8).

[45] https://github.com/MTG/otmm_audio_score_alignment_dataset/tree/2014_fma

[46] http://compmusic.upf.edu/turkish-makam-acapella-sections-dataset

[47] http://compmusic.upf.edu/node/226

3.3 Ontologies

After the corpus generation, I have been working on building ontologies (Gruber, 1995) that formally define the concepts, the properties and the relations in the domain of OTMM.[48] They are based on the existing ontologies such as Friend of a Friend Ontology (**FOAF**) (Brickley & Miller, 2014), *ordered list ontology*[49] and the music ontologies developed by (Raimond, 2008). The entities are organized in four distinct ontologies:

1. **Makam Symbolic Music Ontology**: Based on the *Symbolic Music Ontology*,[50] it defines the entities in the symbolic domain; including symbolic durations (fourth, dotted sixteenth etc.), the accidentals used in classical and folk repertoire (koma, bakiye etc.), stroke names (düm, tek etc.) and traditional note names (gerdaniye, hüseyni etc.).

2. **Makam Ontology**: Defines the OTMM specific entities such as the makams, forms, usuls, chords and çeşnis. This ontology imports the *Makam Symbolic Music ontology*.

3. **Makam Score Ontology**: Defines the concepts related to the music scores such as notes, measures, composers, lyricists, transcribers, compositions. This ontology imports the *Makam ontology*.

4. **OTMM Ontology**: Defines the all entities about OTMM. This ontology imports all the aforamentioned ontologies. In addition, it defines the entities related to performance such as (traditional) instruments, performers, tonic and ahenk.

A visualization of the entities and relations in the ontologies are given in Figure 3.7. I am currently at the stage to generate a knowledge base from the structured (and linked) (meta)data included in the CompMusic OTMM corpus.

[48] https://github.com/sertansenturk/makam-ontologies

[49] http://smiy.sourceforge.net/olo/spec/orderedlistontology.html

[50] http://purl.org/ontology/symbolic-music

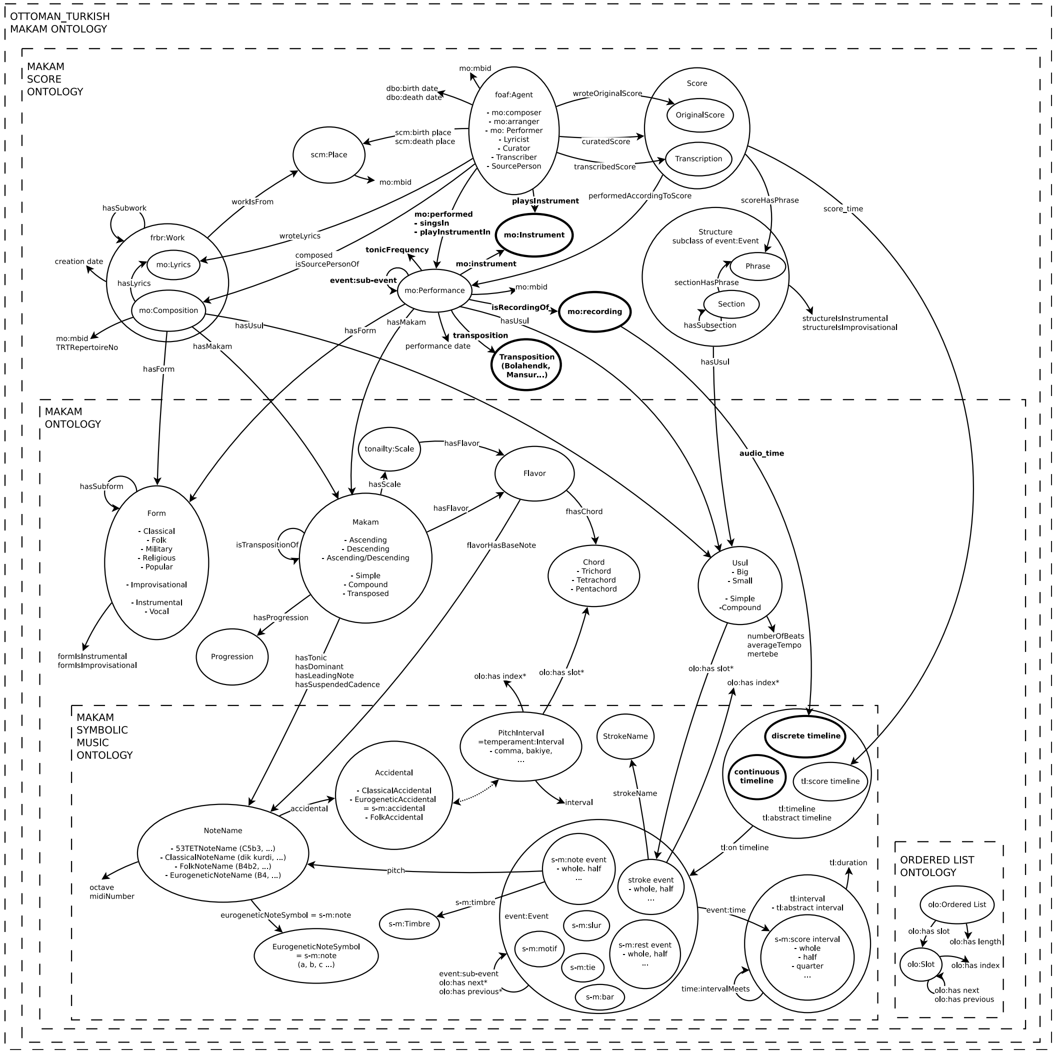


Figure 3.7: A visualization of the entities and interrelationships defined in the OTMM ontologies. The entities and relations in bold are defined in *OTMM ontology*.

3.4 Conclusion

In this Chapter, a research corpus of OTMM is presented. The corpus is created under the considerations to meet some criteria: purpose, quality, completeness, coverage and reusability. We also present some test datasets, which have been used to test and calibrate some computational methodologies, e.g. (Bozkurt, Karaosmanoğlu, et al., 2014; Dzhambazov et al., 2014; Şentürk et al., 2013; Şentürk, Gulati, & Serra, 2014; Şentürk, Holzapfel, & Serra, 2014).

Having created a representative corpus and obtained its automatic description, the next step is to generate a knowledge-base by taking the definitions in the ontologies as the reference. This way, the structured data is linked with each other and other relevant information sources. The resultant linked data would provide us a broader, semantic description of OTMM. Moreover, the ontology specification would allow relevant applications (such as Dunya) to interact with others using common semantics.

The CompMusic OTMM corpus and the relevant test datasets have facilitated most of the research done in the context of this thesis and in the context of computational research applied on OTMM in the recent years. We hope that the CompMusic OTMM corpus will increase both in size and variety and it will continue to stimulate academic studies in MIR and computational musicology in the future.

Chapter 4

Score Analysis

In analyzing a music piece, scores provide an easily accessible symbolic description of many relevant musical components. Moreover they typically include editorial annotations such as the nominal tempo, the rhythmic changes and structural markings. These aspects render the music score a practical source to extract and analyze the melodic, rhythmic and structural properties of the studied music.

In this chapter, the automatic content description process applied to the music scores of OTMM is explained. First, the editorial information in the scores (such as composer, makam and tempo) is parsed. This information is fetched from three different sources, namely the **SymbTr** scores in the txt format, in the mu2 format (the formats are explained in Section 3.1.2) and MusicBrainz. Next, the information is validated against each other. The main contribution in this chapter is the structural analysis methodology (Section 4.3), which aims to extract and label the melodic and lyrics organization both on phrase-level and section-level, using symbolic information available in the music scores of OTMM. The method labels the extracted sections and phrases semiotically according their relations with each other using basic string similarity and graph analysis.

The contributions may be summarized as:

- An automatic structural analysis method applied on Ottoman-Turkish makam music scores.
- A novel semiotic labeling method based on network analysis.

- Automatic description of the **SymbTr** collection encompassing the editorial metadata, the sections and the phrases obtained for 2200 music scores, for more than 1300 music scores and 1750 music scores, respectively.
- An open **SymbTr**-score parser, which fetches the embedded and online metadata and applies the structural analysis method.
- Open source packages for automatic content validation and music format conversion of **SymbTr**-scores, extending the score parser.

The structural analysis and semiotic labeling are published in (Şentürk & Serra, 2016b). The obtained metadata and the structure information is later used in the joint analysis of music scores and audio recordings extensively (Chapter 6).

The structure of the rest of the Chapter is as follows: Section 4.1 presents the parsing and (cross-)validation of the score metadata from **SymbTr**-txt and **SymbTr**-mu2 scores and also from MusicBrainz. Section 4.2 describes the extracted melodic and lyrics features. Section 4.3 explains the structural analysis methodology applied to the **SymbTr**-txt scores. Section 4.5 provides the statistics of the automatic description of the **SymbTr** collection. Section 4.4 demonstrates the additional applications of the score parser and the complementary tools for symbolic score processing. Section 4.6 wraps the Chapter with a brief conclusion and suggestions on future research.

4.1 Metadata

Music scores, typically contain high quality, curated editoral metadata describing the relevant composition or (in case of transcriptions) performance. For this reason music scores, when available, could act as a highly reliable information source for describing the overall characteristics of a music tradition. If the scores are machine-readable, e.g. stored in tab separated values (TSV), extensible markup language (XML) or JSON formats, the basic musical elements may be automatically read and processed by a computer without the need of some sophisticated information retrival techniques. Nevertheless, fetching such metadata may not be straightforward due to several reasons such as the format of the music score (e.g. parsing tables vs. MusicXML), how the data is organized (e.g. parsing a metadata header following a certain schema vs. plain text).

The score-related metadata is extracted from different information sources, namely the **SymbTr**-slug, the **SymbTr**-txt and the **SymbTr**-mu2 of a music score (the formats are explained in Section 3.1.2) and MusicBrainz (Section 3.1.4). The relevant MBID in MusicBrainz is looked up from a <**SymbTr**-slug, MBID> dictionary.[1] The metadata and the information sources are summarized in Table 4.1. Remember that there also exists **SymbTr**-scores, which are transcriptions (Section 3.1.2). The metadata is organized to reflect this relation (Table 4.1).

The metadata obtained from each information source is aggregated to obtain a structured description of the score-related metadata, linked to the relevant data via MusicBrainz.[2] Moreover, the makam, form, usul and tempo obtained from these multiple sources are also cross validated (Section 4.4.3).

4.2 Lyrics and Melody

The lyrics and synthetic melody extracted from each structural element is used to compute the relationships between the structures (Section 4.3). The synthetic melody is also the score feature used for audio-score alignment througout Chapter 6. In section linking experiments (Section 6.2), synthetic HPCPs are computed and compared with synthetic melody. Figure 4.1 shows the lyrics and the synthetic melody (according to the theoretical intervals defined in the AEU theory) and the synthetic HPCPs extracted from an excerpt of the **SymbTr**-score of the composition “Gel Güzelim.”[3]

[1] https://github.com/MTG/SymbTr/blob/v2.4.3/symbTr_mbid.json

[2] https://nbviewer.jupyter.org/github/sertansenturk/symbtrdataextractor/blob/v2.1.0/extractsymbtrdata.ipynb

[3] https://github.com/MTG/SymbTr/blob/v2.4.3/txt/nihavent--sarki--aksak--gel_guzelim--faiz_kapanci.txt

Table 4.1: Summary of the metadata related to a **SymbTr** score. The sources named as “txt, mu2, MB” and “slug” refer to the contents of the **SymbTr**-txt score, the header of the **SymbTr**-mu2 score, MusicBrainz and the **SymbTr**-slug (“[makam]--[form]--[usul]--[title]--[artist]”).

Key	**txt**	**mu2**	**MB**	**slug**	**Explanation**
symbtr				✓	The **SymbTr**-slug, if supplied. Otherwise, the filename
url			✓		The URL of the MusicBrainz attribute (https://musicbrainz.org/work/[mbid] for works, https://musicbrainz.org/recordings/[mbid] for recordings)
work		✓	✓	✓	Related work. If the score is associated with a recording, the key is “works.”
recordings			✓		Recordings related to the work in MusicBrainz. Optional for scores associated with works. If the score is associated with a recording instead of a work, the key is “recording.”
composer		✓	✓	✓	Composer(s) related to the work
lyricist		✓	✓	✓	Lyricist(s) related to the work
makam		✓	✓	✓	Makam of the work
form		✓	✓	✓	Form of the work
usul	✓	✓	✓	✓	Usul of the work
number_of_notes	✓				Number of notes and rests in the **SymbTr**-txt score (annotation rows etc. are not counted)
duration	✓				Duration in nominal tempo
rhythmic_structure	✓				List of rhythmic (e.g. tempo, usul) changes in the score
tempo	✓	✓			Nominal tempo of the piece. Read from the mu2 header, validated with the note durations in the **SymbTr**-txt
notation		✓			Shows whether a score is written in the classical accidentals (“TSM”) or folk accidentals (“THM”).
genre		✓			Indicates whether the composition belongs to the classical or the folk repertoire
key_signature		✓			Validated with the key signature of the makam of the composition, stored in the makam dictionary within the **SymbTr** collection.
language			✓		The lyrics language
scores					The slug of the related **SymbTr**-score, obtained from the <SymbTr-slug, MBID> dictionary in the **SymbTr** collection.
ehtonic				✓	Symbol of the tonic note. Obtained by referring to the karar symbol of the makam of the composition from the makam dictionary in the **SymbTr** collection.
releases			✓		The releases which include the associated recording. Only for the scores, which are associated with recordings instead of works.
artist_credits			✓		The main credited artist(s) of the associated recording. Only for the scores, which are associated with recordings instead of works.
artists			✓		The artist(s), which are related to the associated recording (e.g. performer, conductor). Only for the scores, which are associated with recordings instead of works.

Table 4.2: The features extracted from score fragments and their summary for each computational task they are used as input.

Feature	Source	Task	Frame Rate	Explanation
Lyrics	**SymbTr**-txt	Lyrical relationship computation (Section 4.3.2)	N/A	
Synthetic Melody I	**SymbTr**-txt	Melodic relationship computation (Section 4.3.2)	Least common multiplier of the symbolic durations	
Synthetic Melody II	**SymbTr**-txt	Preliminary section linking experiments (Appendix A)	10ms	The tuning extracted from the audio recording to be aligned is used to generate the synthetic melody.
Synthetic Melody III	**SymbTr**-txt	All audio-score alignment tasks in Chapter 6 except Section 6.8 and Section 6.12	$44100/2048 \approx 46$ms	The rest durations are added to the duration of the previous note (Figure 4.1d). The theoretical tuning defined in the AEU theory is used to generate the synthetic melody.
Synthetic Melody IV	**SymbTr**-txt	Note-Level Alignment (Section 6.8) and Combined Joint Analysis Tasks (Section 6.12)	$44100/2048 \approx 46$ms	Same as Synthetic Melody III, however the melodies are resynthesized with respect to the estimated average tempo (of the recording in the section linking step and of the estimated section in the note-level alignment step).
Synthetic HPCPs	**SymbTr**-MIDI	Section Linking (Section 6.7.4)	$44100/2048 \approx 46$ms	The frame size is chosen as 4096 samples. They are computed with different number of bins per octave in the section linking experiments.

Table 4.2 summarizes the lyrics and melody-related features extracted from the score and their usage in different computational tasks.

4.2.1 Lyrics

The information in the lyrics column is used to determine the boundaries of the vocal sections in Section 4.3.2. The lyrics associated with a sequence or an element x is a string denoted as $\boldsymbol{\lambda}^{(x)}$, simply obtained by contatenating the syllables of the note sequence $\left[\bar{n}_1^{(x)}, \ldots, \bar{n}_{\left|\bar{\mathbf{N}}(x)\right|}^{(x)}\right]$ of x. The editorial annotations (Section 3.1.2) and the whitespaces in the lyrics column are ignored in the process. Then the characters in the obtained string are all converted to lower case. Trivially, $\boldsymbol{\lambda}^{(\bar{n}_i)}$ of a note $\bar{n}_i$ is the syllable associated with the note $\bar{n}_i$ in the lyrics column.

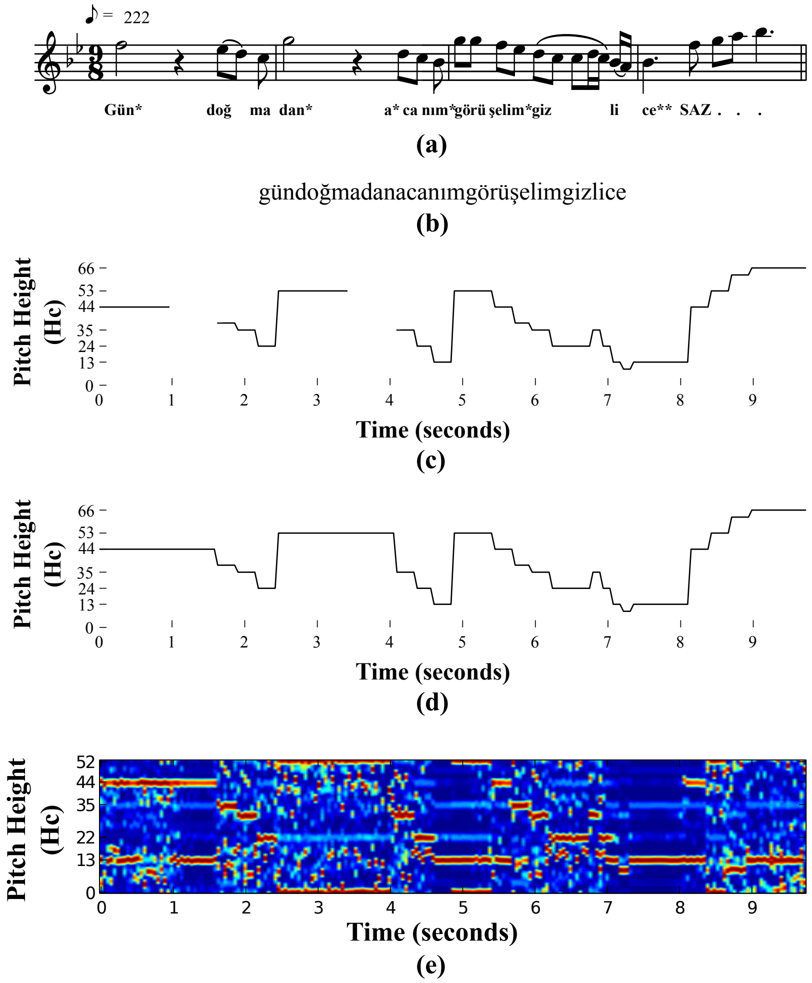
4.2.2 Synthetic Melody

Given a fragment (x) in the **SymbTr**-txt score, the corresponding $\left\langle n_i^{(x)}, d\big(\bar{n}_i^{(x)}\big)\right\rangle$ tuples in the note sequence $\bar{\mathbf{N}}^{(x)}$ is selected. Here, the sum of the durations in the tuples $\sum_i d\big(\bar{n}_i^{(x)}\big)$ is equal to the duration of the score fragment $d(x)$. Then the makam of the composition is noted, which is given in the score, and obtain the karar-symbol of the piece by checking the makam in the <makam,tonic> dictionary (see Table 4.1).

Then the note-symbols $n_i^{(x)}$ are mapped to the Hc distances. The mapping can be done according to a music theory (e.g. the AEU theory) with reference to the karar note. As an example see Figure 4.1: here the karar note is G4 (Nihavent makam) and all the notes take on values in relation to that karar, as for instance 13 Hc for the B4♭. The intervals can also be mapped according to a tuning (Section 5.9) extracted from audio recording(s) (Şentürk, Holzapfel, & Serra, 2012, explained in Appendix A). Finally, a synthetic melody $\hat{\boldsymbol{\Psi}}^{(x)} = \left[\hat{\psi}_1^{(x)}, \dots, \hat{\psi}_{\left|\hat{\boldsymbol{\Psi}}^{(x)}\right|}^{(x)}\right]$ is calculated by sampling the mapped notes relative to the their duration $d\big(\bar{n}_i^{(x)}\big)$ at a certain frame rate and concatenating all samples (Figure 4.1c). In the score structural analysis (Section 4.3.2), the synthetic melody is sampled with a frame rate equal to the least common multiplier (LCM) of the symbolic score durations (e.g. if there are only dotted eighth and fourth notes in a score, the frame rate is a sixteenth note). In the audio-score alignment experiments presented in Chapter 6, the frame rate is selected as $2048/44100 = 46$ms (21.5 samples per second), which is equal to the frame rate of the predominant melody extracted from the audio recordings (Section 6.2).

In makam music practice, the notes preceding rests may be sustained for the duration of the rest.[4] For this reason, the rests in the score may be ignored and their duration may be added to the previous note (Figure 4.1d) in the synthetic melody computation step in the audio-score alignment experiments (Section 6.2).

[4] Figure 6.1 in Chapter 6 shows the same score fragment in Figure 4.1 with a linked audio fragment. Notice that the notes in 6.1a are sustained in the performance as seen in the audio waveform in Figure 6.1b.



(a)

(b)

(c)

(d)

(e)

Figure 4.1: A short excerpt from the score of the composition, *Gel Güzelim*. **a)** The score, **b)** the lyrics, **c)** the synthetic melody computed from the note symbols and durations. The spaces in the end of the syllables are displayed as *s. **d)** the synthetic melody with the rest duration added to the duration of the previous note. **e)** the synthetic HPCPs.

4.2.3 Synthetic Harmonic Pitch Class Profiles

In the section-linking experiments that will be described in Section 6.7.4, **SymbTr** scores in MIDI format are used to obtain the synthetic HPCPs. Given a fragment (x) selected from a MIDI-score, audio is generated from the fragment. TiMidity++[5] is used with the default parameters for the audio synthesis. Since there are no standard SoundFonts of makam music instruments, the default SoundFont is selected.[6] Nevertheless the SoundFont selection should not affect the HPCP computation greatly since HPCPs were reported to be robust to changes in timbre (Gómez, 2006). Then, the synthetic HPCPs, $\hat{\mathbf{\Omega}}^{(x)} = \left[\hat{\omega}_1^{(x)}, \ldots, \hat{\omega}_{\left|\hat{\mathbf{\Omega}}^{(x)}\right|}^{(x)}\right]$, are computed for the score fragment (x) (Figure 6.1e). The HPCPs are extracted using Essentia (Bogdanov et al., 2013) with the default parameters given in (Gómez, 2006). The hop size and the frame size are chosen to be 2048 (e.g. ~ 21.5 frames per second) and 4096 samples respectively. The first bin of the HPCPs is assigned to the tonic symbol, $\kappa^{(x)}$, (e.g.G4 for Nihavent makam). Note that the HPCPs contain microtonal information as well, since this information is encoded into the **SymbTr**-MIDI scores (Karaosmanoğlu, 2012). HPCPs, $\hat{\mathbf{\Omega}}^{(x)}$, are computed with different number of bins per octave in the section linking experiments (see Section 6.7.4).

4.3 Structure

Analyzing the structure of a music piece is integral in understanding how the musical events progress along with their functionality within the piece. Automatic extraction of the melodic and lyrics structures, as well as their roles within the composition, might be used to facilitate and enhance tasks such as digital music engraving, automatic form identification and analysis, audio-score and audio-lyrics alignment, music prediction and generation.

Structural analysis is a complex problem, which can be approached in different granularities such as sections, phrases and motifs (Pearce, Müllensiefen, & Wiggins, 2010). To find such groupings there has been many approaches based on music theory (Jackendoff, 1985), psychological findings and computational models (Cambouropoulos, 2001; Pearce et al., 2010). On the other hand, there are a few studies that has investigated automatic structural analysis of makam musics. Lartillot and Ayari (2009) has

[5] `http://timidity.sourceforge.net/`

[6] grand acoustic piano: `http://freepats.zenvoid.org/sf2/`

used computational models to segment Tunisian modal music and compared the segmentations with the annotations of the experts. Later, Lartillot, Yazıcı, and Mungan (2013) has proposed a similar segmentation model for OTMM and also conducted comparative experiments between the automatic segmentations and human annotations. Due to the lack of musicological agreement on how to segment makam music scores, Bozkurt, Karaosmanoğlu, et al. (2014) focused on learning a model from a dataset of music scores annotated by experts and segmenting larger score datasets automatically using the learned model. They propose two novel culture-specific features based on the melodic and rhythmic properties of OTMM and conduct comparative studies with the features used in the state-of-the-art methods (Bod, 2002; Cambouropoulos, 2001; Temperley, 2004; Tenney & Polansky, 1980) and show that the proposed features improve the phrase segmentation performance.[7] These methods typically focus on finding the segment boundaries and do not study the inter-relations between the extracted segments.

4.3.1 Problem Definition

Given the note sequence $\bar{\mathbf{N}}^{(b)} := \left[\bar{n}_1^{(b)}, \bar{n}_2^{(b)}, \ldots\right]$ and the measure sequence $\bar{\mathbf{M}}^{(b)} := \left[\bar{m}_1^{(b)}, \bar{m}_2^{(b)}, \ldots\right]$ in the score (b), the aim is to extract the sections $\bar{\mathbf{S}}^{(b)} := \left[\bar{s}_1^{(b)}, \bar{s}_2^{(b)}, \ldots\right]$ and the phrases $\bar{\mathbf{P}}^{(b)} := \left[\bar{p}_1^{(b)}, \bar{p}_2^{(b)}, \ldots\right]$ along with their boundaries, and the melodic and lyrics relationship with other structural elements of the same type. Note that the sections and phrases will be collectively called as "structural elements" in the score throughout the Chapter. Moreover, the superscript (b) is omitted in the rest of the Chapter for the sake of simplicity.

It is assumed that the structural elements of the same type are non-overlapping and consecutive (e.g. the last note of a section is always adjacent to the first note of the next section). Consecutiveness restriction also implies that transitive interactions between two

[7] For a detailed review of structural analysis applied to OTMM and relevant state of the art the readers are referred to (Bozkurt, Karaosmanoğlu, et al., 2014) and (Pearce et al., 2010), respectively.

consecutive structural elements are not permitted. In the scores of the vocal compositions, each poetic line is considered as a section.

Remark that each subsequence[8] might cover or overlap with subsequences of different types, e.g. the note sequence in a section would be a subsequence of $\bar{\mathbf{N}}$ or a phrase might start in the middle of a measure and end in another. The index of the first note and the index of the last note in the note sequence $\bar{\mathbf{N}}^{(x)}$ of a score element x are denoted as $\bar{n}_1^{(x)}$ and $\bar{n}_{\left|\bar{\mathbf{N}}^{(x)}\right|}^{(x)}$ (where $\left|\bar{\mathbf{N}}^{(x)}\right|$ is the number of notes in $\bar{\mathbf{N}}^{(x)}$), respectively. For example, the first note of an arbitrary section $\bar{s}_i$, phrase $\bar{p}_j$ and measure $\bar{m}_k$ are denoted as $\bar{n}_1^{(\bar{s}_i)}$, $\bar{n}_1^{(\bar{p}_j)}$ and $\bar{n}_1^{(\bar{m}_k)}$, respectively. The lyrics associated with an arbitrary element x is denoted as $\boldsymbol{\lambda}^{(x)}$. Each $\bar{n}_j^{(b)}$ (where $j \in \left[1 : |\bar{\mathbf{N}}^{(b)}|\right]$) consists of a $\left\langle j, n_j^{(b)}, d\left(n_j^{(b)}\right), \boldsymbol{\lambda}^{\left(n_j^{(b)}\right)} \right\rangle$ tuple, the elements of which represent the note index, the note symbol, the note duration and the syllable in the lyrics associated with the note.

4.3.2 Methodology

First, the section boundaries are extracted from the score using a *ad hoc*, heuristic process taking the editorial structure labels in the score as an initial reference. In parallel, the score is automatically segmented phrases according to a model learned from the phrases annotated by an expert. Next, the synthetic melody and the lyrics are extracted from each section and phrase. Then, a melodic and a lyrics similarity matrix are computed between the extracted phrases and the sections separately. A graph is formed from each similarity matrix and the relation between the structural elements in the context of the similarity (melodic or lyrics) is obtained. Finally semiotic labeling is applied to the computed relations (Şentürk & Serra, 2016b).

Section Extraction

The section boundaries are inferred using the explicit and implicit boundaries given in the lyrics column of the **SymbTr**-txt scores

[8] or element, which can also be regarded as a subsequence composed of a single element.

(Section 3.1.2). As a preprocessing step to distinguish the instrumental section labels from other editorial annotations in the lyrics column, the unique strings are extracted in the lyrics column of all **SymbTr** scores. The set of editorial annotations in the **SymbTr**-scores is basically the set of the strings written in capital letters. Then, the section annotations are picked manually from the set of editorial annotations.[9]

Given the score (b), the set of instrumental section names are searched in the lyrics column. The matched note indices mark the actual beginning $\bar{n}_1^{(\bar{s}_i)}$s of the instrumental sections $\bar{s}_i \in \bar{\mathbf{S}} \,|\, \boldsymbol{\lambda}^{(\bar{s}_i)} = \varnothing$. Next, the lyrics column is searched for syllables ending with double spaces. The index of the matched notes are assigned to the final note $\bar{n}_{\left|\bar{\mathbf{N}}^{(\bar{s}_i)}\right|}^{(\bar{s}_i)}$s of the vocal sections; $\bar{s}_i \in \bar{\mathbf{S}} \,|\, \boldsymbol{\lambda}^{(\bar{s}_i)} \neq \varnothing$. As explained in Section 3.1.2, the indices $\bar{n}_{\left|\bar{\mathbf{N}}^{(\bar{s}_i)}\right|}^{(\bar{s}_i)}$s may not coincide with the actual ending and it may be moved to a subsequent note.

Up to here, the section sequence $\bar{\mathbf{S}} := [\bar{s}_1, \bar{s}_2, \dots, \bar{s}_{|\bar{\mathbf{S}}|}]$ has been found, where $|\bar{\mathbf{S}}|$ is the total number of sections in the music score (b). The first note of the vocal sections and the last note of the instrumental sections are unassigned at this stage. The section boundaries are located using a rule-based scheme iterating though all sections starting from the last one.

If a section $\bar{s}_i$ is instrumental, then $\bar{n}_1^{(\bar{s}_i)}$ is already assigned. If a section $\bar{s}_i$ is vocal and the previous section $\bar{s}_{i-1}$ is instrumental, the last instrumental measure is found, $\bar{m}_k \in \bar{\mathbf{M}} \,|\, \boldsymbol{\lambda}^{(\bar{m}_k)} = \varnothing$, before the last note $\bar{n}_{\left|\bar{\mathbf{N}}^{(\bar{s}_i)}\right|}^{(\bar{s}_i)}$ of the section $\bar{s}_i$. Then, the first note $\bar{n}_1^{(\bar{s}_i)}$ is assigned to the first note $\bar{n}_1^{(\bar{m}_{k+1})}$ of the next measure $\bar{m}_{k+1}$. If both the current section $\bar{s}_i$ and the previous section $\bar{s}_{i-1}$ are vocal, $\bar{n}_1^{(\bar{s}_i)}$ is assigned to the index of the first note with lyrics after the last note $\bar{n}_{\left|\bar{\mathbf{N}}^{(\bar{s}_{i-1})}\right|}^{(\bar{s}_{i-1})}$ of $\bar{s}_{i-1}$. If $\bar{n}_1^{(\bar{s}_i)}$ and $\bar{n}_{\left|\bar{\mathbf{N}}^{(\bar{s}_{i-1})}\right|}^{(\bar{s}_{i-1})}$ are not in the same measure, $\bar{n}_1^{(\bar{s}_i)}$ is reassigned to the first note of its measure, i.e. $\bar{n}_1^{(\bar{m}_k)} \mid \bar{n}_1^{(\bar{s}_i)} \in \bar{m}_k$. Finally the last note $\bar{n}_{\left|\bar{\mathbf{N}}^{(\bar{s}_i)}\right|}^{(\bar{s}_i)}$ of the section is moved to the index of the first note $\bar{n}_1^{(\bar{s}_{i+1})}$ of the next section $\bar{s}_{i+1}$ minus one.

[9] https://github.com/sertansenturk/symbtrdataextractor/blob/v2.1.0/symbtrdataextractor/makam_data/symbTrLabels.json

The pseudocode of the procedure is given in Algorithm 1. Note that the start of the first section and the end of the final section are assigned to 1 and $|N|$, respectively, where $|N|$ is the number of notes in the score. This detail omitted from the pseudocode for the sake of brevity.

Algorithm 1 Locate section boundaries

```
for $i := |\bar{\mathbf{S}}| \to 1$ do                                ▷ from the last index to the first
    if $\boldsymbol{\lambda}^{(\bar{s}_i)} \neq \varnothing$ then                       ▷ find the start of the vocal section
        if $\boldsymbol{\lambda}^{(\bar{s}_{i-1})} = \varnothing$ then                ▷ previous section is instrumental
            $k \leftarrow arg_k min\big(\bar{n}_1^{(\bar{m}_k)}$ is after $\bar{n}_1^{(\bar{s}_{i-1})} \wedge$
                        $\boldsymbol{\lambda}^{(\bar{m}_k)} \neq \varnothing\big)$
            $\bar{n}_1^{(\bar{s}_i)} \leftarrow \bar{n}_1^{(\bar{m}_k)}$
        else                                                   ▷ previous section is vocal
            $k \leftarrow arg_k min\big(n_k$ is after $\bar{n}^{(\bar{s}_{i-1})}_{|\bar{\mathbf{N}}^{(\bar{s}_{i-1})}|} \wedge$
                        $\boldsymbol{\lambda}^{(\bar{n}_k)} \neq \varnothing\big)$
            if $\bar{n}_1(\bar{s}_i) \in \bar{m}_k \ \wedge \ \bar{n}^{(\bar{s}_{i-1})}_{|\bar{\mathbf{N}}^{(\bar{s}_{i-1})}|} \notin \bar{m}_k$ then
                $\bar{n}_1^{(\bar{s}_i)} \leftarrow \bar{n}_1^{(\bar{m}_k)}$
            else
                $\bar{n}_1^{(\bar{s}_i)} \leftarrow n_k$
    $\bar{n}^{(\bar{s}_i)}_{|\bar{\mathbf{N}}^{(\bar{s}_i)}|} \leftarrow \bar{n}_1^{(\bar{s}_{i+1})} - 1$                            ▷ sections are consecutive
```

Having located the boundaries, the sections are extracted by simply taking all information (i.e. rows in the **SymbTr**-txt score) between these note boundaries. Figure 4.2 shows the section boundaries obtained on a mock example.

Automatic Phrase Segmentation

We use the automatic phrase segmentation methodology, which is proposed by Bozkurt, Karaosmanoğlu, et al. (2014). The source code and the training dataset presented in (Karaosmanoğlu et al., 2014) are open and available online.[10]

In order to train the segmentation model, the annotations of Expert 1, who annotated all the 488 scores in the training dataset, are

[10] `http://www.rhythmos.org/shareddata/turkishphrases.html`

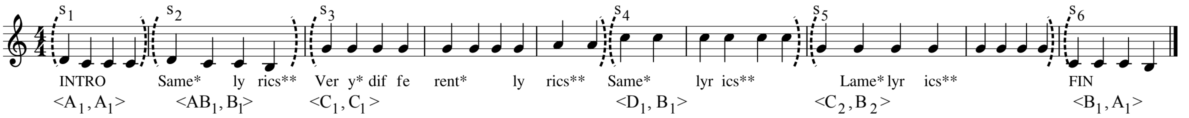


Figure 4.2: Section analysis applied to a mock example. The section labels ("INTRO" and "FIN") are given in the lyrics written in capital letters, The spaces in the end of the syllables are visualized as *. The semiotic $< \mathit{Melody}, \mathit{Lyrics} >$ label tuples of each section are shown below the lyrics. The similarity threshold in the similar clique computation step is selected as 0.7 for both melody and lyrics.

used (Karaosmanoğlu et al., 2014). Moreover, the authors of (Karaosmanoğlu et al., 2014) commented through personal communication that the first expert's annotations are more consistent with each other. There are a total of 20801 training phrases annotated by the first expert. In the training dataset, the variants of some usuls are combined (e.g. the scores in Kapalı curcuna usul is treated as Curcuna. Using the trained model, automatic phrase segmentation is applied to the score collection (Section 3.1.2) and the phrase boundaries $\bar{n}_1^{(\bar{p}_k)}$ and $\bar{n}_{|\bar{\mathbf{N}}^{(\bar{p}_k)}|}^{(\bar{p}_k)}$ for each phrase $\bar{p}_k \in P := [\bar{p}_1, \bar{p}_2, \dots]$ is obtained, where P is the automatically extracted phrase sequence. In Figure 4.4, the vertical red and purple lines shows the phrase boundaries extracted from the score "Kimseye Etmem Şikayet."[11]

Melodic and Lyrical relationship computation

Given the structure sequence $F := [f_1, f_2, \dots]$ (which is either the section sequence $\bar{\mathbf{S}}$ or the phrase sequence $\bar{\mathbf{P}}$) extracted from the score, first the synthetic melody is generated and the lyrics of each structural element is extracted (Section 4.2.1). The sampling rate of the synthetic melody is taken as the LCM of the symbolic score durations (e.g. if there are only dotted eighth and fourth notes in a score, the sampling rate is a sixteenth note). This sampling rate allows to represent the melody discretely without introducing any ambiguity in time using the least number of samples possible.

[11] https://github.com/MTG/SymbTr/blob/a50a16ab4aa2f30a278611f333ac446737c5a877/txt/nihavent--sarki--kapali_curcuna--kimseye_etmem--kemani_sarkis_efendi.txt

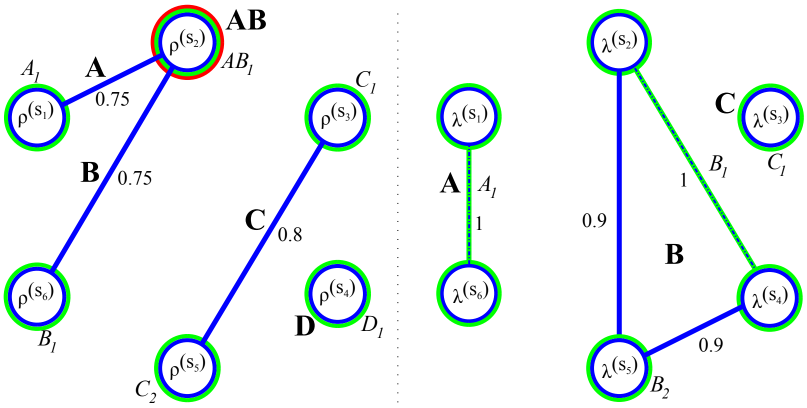
Then, a melodic similarity and lyrics similarity between each element is computed using a similarity measure based on Levenshtein distance (Levenshtein, 1966). The similarity measure $\hat{\mathscr{L}}(x, y)$ is defined as:

$$\hat{\mathscr{L}}(x, y) := 1 - \frac{\mathscr{L}(x, y)}{max\left(|x|, |y|\right)} \tag{4.1}$$

where $\mathscr{L}(x, y)$ denotes the Levenshtein distance between the two "strings" x and y with the lengths $|x|$ and $|y|$, respectively and $max()$ denotes the maximum operation. In this context, x and y are either the synthetic melody or the lyrics of two structural elements. The similarity yields a result between 0 and 1. If the strings of the compared structural elements are exactly the same, the similarity equals to 1. Similar strings (e.g. the melodies of two instances of the same section with volta brackets) would also output a high similarity.

From the melodic and lyrics similarities, two separate graphs are constructed, in which the nodes are the structural elements and the elements are connected to each other with undirected edges. The weight of an edge connecting two structural elements f_i and f_j is equal to $\hat{\mathscr{L}}\left(\hat{\mathbf{\Psi}}^{(f_i)}, \hat{\mathbf{\Psi}}^{(f_j)}\right)$ in the melodic relation graph and $\hat{\mathscr{L}}\left(\boldsymbol{\lambda}^{(f_i)},\ \boldsymbol{\lambda}^{(f_j)}\right)$ in the lyrics relation graph, respectively. Next, the edges are removed with a weight less than a constant similarity threshold $l \in [0, 1]$. In Section 4.3.3, the effect of using different l values will be experimented.

Given the graph, the structural elements with similar strings are grouped by finding the maximal cliques in the graph (Tomita, Tanaka, & Takahashi, 2006). A maximal clique is a subgraph, which has its each node connected to each other and it cannot be extended by including another node. These cliques are denoted as $v_j \in \boldsymbol{\mathcal{V}}$, where $\boldsymbol{\mathcal{V}}$ is the set of "similar cliques." The maximal cliques of the graph are additionally computed only considering the edges with zero weight. These cliques show the groups of structural elements, which have exactly the same string. Each of these cliques are called as "unique clique" $u_k \in \boldsymbol{\mathcal{U}}$, where $\boldsymbol{\mathcal{U}}$ is the set of the "unique clique." Note that two or more similar cliques can intersect with each other. Such an intersection resembles all the relevant similar cliques. These "intersections" are de-



(a) (b)

Figure 4.3: The graphs, the cliques and the semiotic labels obtained from the mock example (Figure 4.2) using an edge weight threshold of 0.7 for both melody and lyrics. The circles represent the nodes and the lines represent the edges of the graphs, respectively. The edge weights are shown next to the lines. Green, blue and red colors represent the unique cliques, the similar cliques and the intersection of similar cliques, respectively. The semiotic label of each similar clique and each intersection is shown in bold and the semiotic label of each unique clique is shown in italic, respectively.

noted as $w_l \in \boldsymbol{\mathcal{W}}$, where $\boldsymbol{\mathcal{W}}$ is the set of intersections between different similar cliques. Also, $\boldsymbol{\mathcal{N}}(\boldsymbol{\mathcal{G}})$ denotes the nodes of an arbitrary graph $\boldsymbol{\mathcal{G}}$. Remark that:

- A unique clique is a subgraph of at least one similar clique, i.e. $\forall u_k \in \boldsymbol{\mathcal{U}}$, $\exists v_j \in \boldsymbol{\mathcal{V}} \mid \boldsymbol{\mathcal{N}}(u_k) \subseteq \boldsymbol{\mathcal{N}}(v_j)$.
- A unique clique cannot be a subgraph of more than one intersection, i.e. $\forall u_k \in \boldsymbol{\mathcal{U}}$, $\nexists\{w_l, w_m\} \subseteq \boldsymbol{\mathcal{W}} \mid \boldsymbol{\mathcal{N}}(u_k) \subseteq \boldsymbol{\mathcal{N}}(w_l) \wedge \boldsymbol{\mathcal{N}}(u_k) \subseteq \boldsymbol{\mathcal{N}}(w_m)$.
- A structural element belongs to only a single unique clique, i.e. $\forall f_i \in F$, $\exists! u_k \in \boldsymbol{\mathcal{U}} \mid \boldsymbol{\mathcal{N}}(f_i) \subseteq \boldsymbol{\mathcal{N}}(u_k)$.

Figure 4.3 shows the graphs computed from the sections of the mock example introduced in Figure 4.2. In the melodic relations graph, each section forms a unique clique since the melody of each section is not exactly the same with each other. Using a similiarty threshold of 0.7, four similar cliques are obtained formed by

$\{\bar{s}_1, \bar{s}_2\}$, $\{\bar{s}_2, \bar{s}_6\}$, $\{\bar{s}_3, \bar{s}_5\}$, $\{\bar{s}_4\}$. Notice that $\{\bar{s}_4\}$ is not connected to any clique. so it forms both a unique clique and a similar clique. Moreover, $\bar{s}_2$ is a member of both the first and the second similar cliques and hence it is the intersection of these two cliques. For the lyrics, there are four unique cliques, formed by the sections $\{\bar{s}_1, \bar{s}_6\}$ (aka. instrumental sections), $\{\bar{s}_2, \bar{s}_4\}$, $\{\bar{s}_3\}$ and $\{\bar{s}_5\}$. The lyrics of $\bar{s}_5$ is very similar to $\{\bar{s}_2, \bar{s}_4\}$ and they form a similar clique composed of these three nodes and the relevant edges.

Semiotic Labeling

After forming the cliques, semiotic labeling explained in (Bimbot, Deruty, Sargent, & Vincent, 2012) is used to describe the structural elements. First similar cliques are labeled with a base letter ("A", "B", "C", …). Then the intersections are labeled by concetanating the base letters of the relevant similar cliques (e.g. "AB", "BDE", …). Each unique clique is finally labeled with the label of the relevant intersection, if exists, or with the label of the relevant similar clique, plus a number according to the order of occurence of the clique in the score. Right now, only the simple labels termed by Bimbot et al. (2012) (e.g. "A_1", "A_2", "AB_2") are used to label the unique cliques.

The pseudocode of the process is given in Algorithm 2. During labeling, $\boldsymbol{\mathcal{U}}$, $\boldsymbol{\mathcal{V}}$ and $\boldsymbol{\mathcal{W}}$ are enumerated by sorting the elements with respect to the index of the first occurence each element in the score. The semiotic melody and lyrics label of an arbitrary element x are denoted as $\Lambda_{mel}^{(x)}$ and $\Lambda_{lyr}^{(x)}$, respectively. In the algorithm, the iterators $\#(v_j)$ for each similar clique v_j and $\#(w_l)$ for each each intersection w_l are used to assign the numerical index to each unique clique $u_k \in \boldsymbol{\mathcal{U}}$ according its relation with the relevant similar clique or intersection.

The label of each section of the mock example is shown below the staff in Figure 4.2. The same semiotic labels are also shown on the computed graphs in Figure 4.3. Notice that the melodic semiotic label of $\bar{s}_6$ is B_1 because the first occurence of the relevant similar clique is at $\bar{s}_2$.

By extracting the relations in the graphs computed from the melodic and lyrics similarity matrices (Section 4.3.2) and then applying semiotic labeling to each section and phrase according to

Algorithm 2 Semiotic labeling

```
$@ \leftarrow$“$A$”    ▷ Start the base letter iterator from “$A$”
$\#(v_j) \leftarrow 1, \forall v_j \in \boldsymbol{\mathcal{V}}$    ▷ Initialize the num. iterators for all $v_j$
$\#(w_l) \leftarrow 1, \forall w_l \in \boldsymbol{\mathcal{W}}$    ▷ Initialize the num. iterators for all $w_l$
for $v_j \in sort(\boldsymbol{\mathcal{V}})$ do    ▷ Label similar cliques
    $\Lambda^{(v_j)} \leftarrow @$
    increment $@$    ▷ “$A$” ⇒ “$B$”
for $w_l \in sort(\boldsymbol{\mathcal{W}})$ do    ▷ Label intersections
    $\Lambda^{(w_l)} \leftarrow$ concatenate $\Lambda^{(v_j)}, \forall(v_j) \mid \boldsymbol{\mathcal{N}}(w_l) \subseteq \boldsymbol{\mathcal{N}}(v_j)$
for $u_k \in sort(\boldsymbol{\mathcal{U}})$ do    ▷ Label unique cliques
    if $\exists w_l \mid \boldsymbol{\mathcal{N}}(u_k) \subseteq \boldsymbol{\mathcal{N}}(w_l)$ then
        $\Lambda^{(u_k)} \leftarrow \Lambda^{(w_l)}_{\#(w_l)}$    ▷ e.g. “$ACD_1$”
        $\#(w_l) \leftarrow \#(w_l) + 1$
    else
        $\Lambda^{(u_k)} \leftarrow \Lambda^{(v_j)}_{\#(v_j)} \mid \boldsymbol{\mathcal{N}}(u_k) \subseteq \boldsymbol{\mathcal{N}}(v_j)$    ▷ e.g. “$C_2$”
        $\#(v_j) \leftarrow \#(v_j) + 1$
for $f_i \in F$ do    ▷ Label structural elements
    $\Lambda^{(f_i)} \leftarrow \Lambda^{(u_k)} \mid \boldsymbol{\mathcal{N}}(f_i) \subseteq \boldsymbol{\mathcal{N}}(u_k)$
```

its relation, a $< \Lambda_{mel}, \Lambda_{lyr} >$ tuple is obtained for each section and phrase (Section 4.3.2). For each phrase the sections are also marked, which enclose and/or overlap with the phrase.

Figure 4.4 shows the results of the structural analysis applied to the score “Kimseye Etmem Şikayet.” The sections are displayed in colored boxes with the volta brackets colored with a darker shade of the same color. The section labels and their semiotic $< \Lambda_{mel}, \Lambda_{lyr} >$ label tuple is shown on the left. The phrase boundaries are shown as red lines for the first and as purple for the second pass. The phrases and their semiotic labels are shown on top of the relevant interval and on the bottom, when there are differences in the boundaries in the second pass. Note that $\bar{s}_5, \bar{s}_6, \bar{s}_9$ and $\bar{s}_{10}$ are the repetitive poetic lines (tr: “Nakarat”). “[Son]” in the end of the “Nakarat” marks the end of the piece. The similarity threshold is taken as 0.7 for **a)** melody and **b)** lyrics. The usul of the score is Kapalı curcuna, which is treated as Curcuna in the phrase segmentation step (Section 4.3.2). Further examination of the analysis is left to the

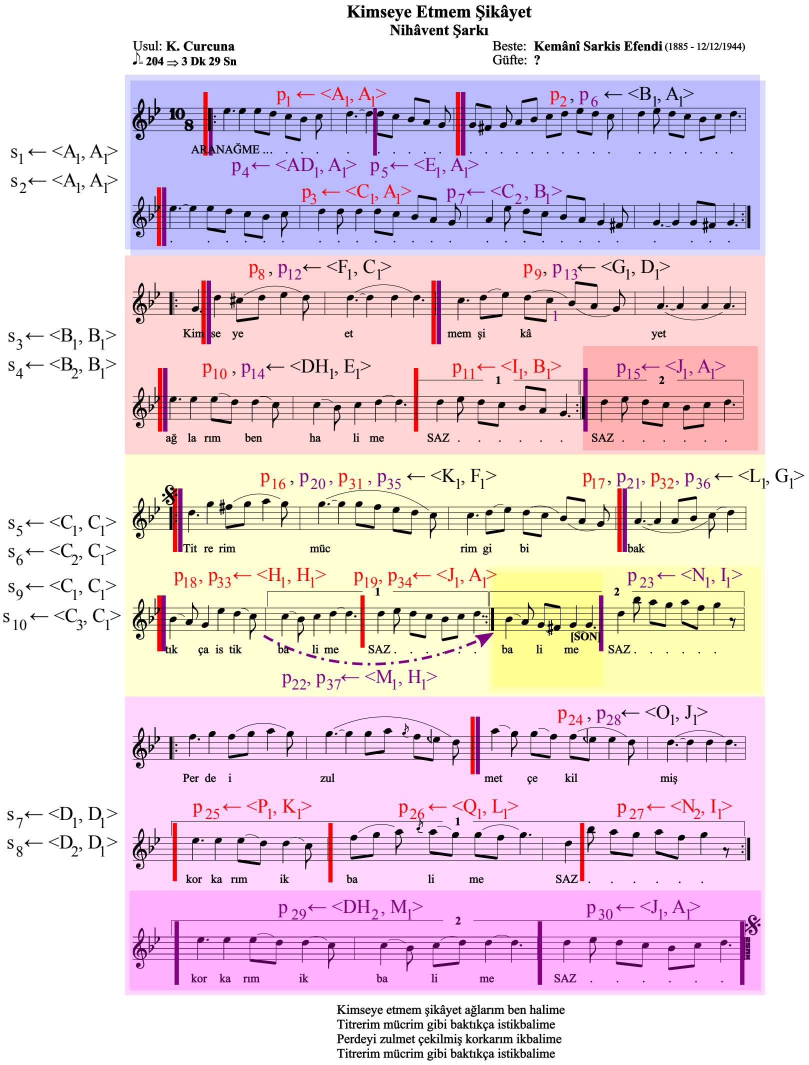

Kimseye Etmem Şikâyet
Nihâvent Şarkı
Usul: K. Curcuna
♪= 204 ⇒ 3 Dk 29 Sn
Beste: Kemânî Sarkis Efendi (1885 - 12/12/1944)
Güfte: ?
$s_1 \leftarrow <A_1, A_1>$
$s_2 \leftarrow <A_1, A_1>$
$p_1 \leftarrow <A_1, A_1>$
$p_2, p_6 \leftarrow <B_1, A_1>$
ARANAĞME ...
$p_4 \leftarrow <AD_1, A_1>$
$p_5 \leftarrow <E_1, A_1>$
$p_3 \leftarrow <C_1, A_1>$
$p_7 \leftarrow <C_2, B_1>$
$s_3 \leftarrow <B_1, B_1>$
$s_4 \leftarrow <B_2, B_1>$
$p_8, p_{12} \leftarrow <F_1, C_1>$
$p_9, p_{13} \leftarrow <G_1, D_1>$
Kim se ye et mem şi kâ yet
$p_{10}, p_{14} \leftarrow <DH_1, E_1>$
$p_{11} \leftarrow <I_1, B_1>$
$p_{15} \leftarrow <J_1, A_1>$
ağ la rım ben ha li me SAZ . . . . SAZ . . . .
$s_5 \leftarrow <C_1, C_1>$
$s_6 \leftarrow <C_2, C_1>$
$s_9 \leftarrow <C_1, C_1>$
$s_{10} \leftarrow <C_3, C_1>$
$p_{16}, p_{20}, p_{31}, p_{35} \leftarrow <K_1, F_1>$
$p_{17}, p_{21}, p_{32}, p_{36} \leftarrow <L_1, G_1>$
Tit re rim müc rim gi bi bak
$p_{18}, p_{33} \leftarrow <H_1, H_1>$
$p_{19}, p_{34} \leftarrow <J_1, A_1>$
$p_{23} \leftarrow <N_1, I_1>$
tık ça is tik ba li me SAZ . . . . ba li me [SON] SAZ . . . .
$p_{22}, p_{37} \leftarrow <M_1, H_1>$
$p_{24}, p_{28} \leftarrow <O_1, J_1>$
Per de i zul met çe kil miş
$s_7 \leftarrow <D_1, D_1>$
$s_8 \leftarrow <D_2, D_1>$
$p_{25} \leftarrow <P_1, K_1>$
$p_{26} \leftarrow <Q_1, L_1>$
$p_{27} \leftarrow <N_2, I_1>$
kor ka rım ik ba li me SAZ . . . .
$p_{29} \leftarrow <DH_2, M_1>$
$p_{30} \leftarrow <J_1, A_1>$
kor ka rım ik ba li me SAZ . . . .
Kimseye etmem şikâyet ağlarım ben halime
Titrerim mücrim gibi baktıkça istikbalime
Perdeyi zulmet çekilmiş korkarım ikbalime
Titrerim mücrim gibi baktıkça istikbalime

Figure 4.4: The results of the automatic structural analysis of the score "Kimseye Etmem Şikayet."

readers as an exercise.

4.3.3 Experiments

Bozkurt, Karaosmanoğlu, et al. (2014) report the evaluation of the phrase segmentation method described in Section 4.3.2 on an earlier and slightly smaller version of the training dataset. The readers are referred to Bozkurt, Karaosmanoğlu, et al. (2014) for the evaluation of the training data. Furthermore, the labels of the automatic phrase segmentations need to be validated by musicologists parallel to the discussions brought by (Bozkurt, Karaosmanoğlu, et al., 2014). For this reason, investigation of the effects of the similarity threshold l in phrase analysis is left as future research.

To observe the effect of the similarity threshold in the melodic and lyrics relationship extraction (Section 4.3.2), a small dataset from the **SymbTr** collection is collected. The test dataset consists of 23 vocal compositions in the şarkı form and 42 instrumental compositions in peşrev and sazsemaisi forms. These three forms are the most common forms of the classical OTMM repertoire. Moreover their sections are well-defined within the music theory; the two instrumental forms typically consists of four distinct Hanes and a Teslim section, which follow a verse-refrain-like structure; the sections of the şarkıs typically coincide with the poetic lines. The experiments on şarkıs are focused on the ones with the poetic organization "Zemin, Nakarat, Meyan, Nakarat," which is one of the most common poetic organization observed in the şarkı form. Using the automatically extracted section boundaries (Section 4.3.2) as the ground-truth, I have manually labeled the sections in the scores with the same naming convention explained in Section 4.3.2.[12] Due to lack of data and concerns regarding subjectivity, the evaluation of section boundaries is left as future research.

Section analysis experiments are conducted on the test dataset by varying the similarity threshold from 0 to 1 with a step size of 0.05. After the section labels are obtained, the semiotic melody and lyrics labels are compared with the annotated labels. An automatic label is considered as "True," if it is exactly the same with the annotated label and "False," otherwise. For each score, the labeling accuracy is computed for the melody and the lyrics separately by dividing number of correctly identified labels with the total number of sections. Additionally, the number of similar cliques and its ratio to the unique cliques obtained for each score is noted. For each experiment, the average accuracy for the similarity threshold

[12] The experiments and results are available at `https://github.com/sertansenturk/otmm-score-structure-experiments/releases/tag/fma_2016`

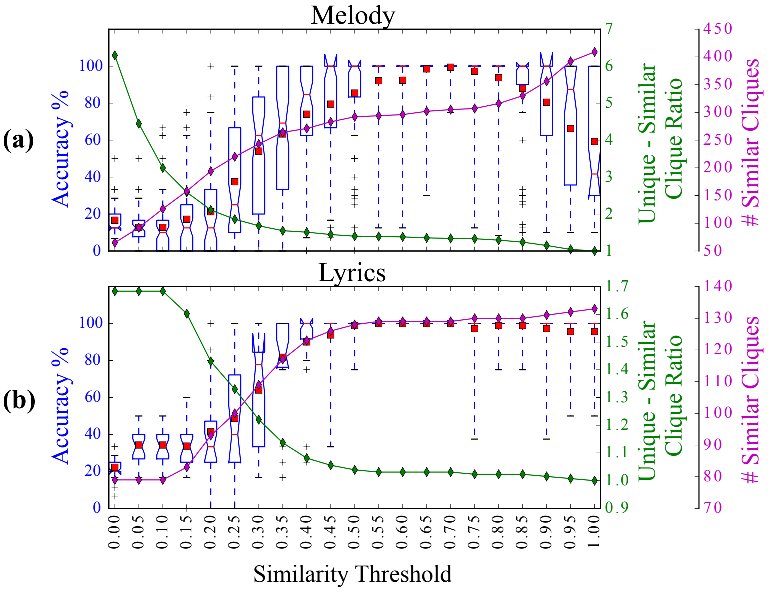


Figure 4.5: The notched boxplots of the accuracies, number of similar cliques and the ratio between the number of unique cliques and similar cliques obtained for **a)** the melody labels and **b)** the lyrics labels (only for vocal compositions) using different similarity thresholds. The squares in the boxplots denote the mean accuracy.

l is computed by taking the mean of the accuracies obtained from each score.

Figure 4.5 shows the notched boxplots of the accuracies, the total number of similar cliques and the ratio between the number of unique cliques and the number of similar cliques obtained for each similarity threshold. For the melody labels, the best results are obtained for the similarity threshold values between 0.55 and 0.80 and the best accuracy is 99%, when l is selected as 0.70. For lyrics labeling, any similarity value above 0.35 yields near perfect results and 100% accuracy is obtained for all the values of l between 0.55 and 0.70. In parallel, the number of similar cliques and the ratio between the unique cliques and the similar cliques gets flat in these regions. From these results, the optimal l as 0.70 is selected for both melodic and lyrics similarity.

4.3.4 Discussion

As shown in Section 4.3.3, the similarity threshold l has a direct impact on the structure labels. A high threshold might cause most of the similar structural elements regarded as different, whereas a low threshold would result in many differences in the structure disregarded. In this sense the extreme values of l (around 0 or 1), would not provide any meaningful information as $l = 0$ would result in all the structures being labeled similar and $l = 1$ would be output all the structures as unique. It is also observed that the melodic similarity is more sensitive to value of l than lyrics similarity. This is expected as the strings that make up the lyrics are typically more diverse than the note symbols used to generate the synthetic melody. In the experiments the optimal value of l is found as 0.7 for the small score dataset of compositions in the peşrev, sazsemaisi and şarkı forms. Moreover, it is observed that the curves representing the number of similar cliques and the ratio between the unique cliques and the similar cliques are relatively flat around the same l value, where obtain the best results are obtained (Figure 4.5). This implies that there is a correlation between decisions of the annotator and the methodology.

Nevertheless, the optimal l value presented above should not be considered as a general optimal. First of all, the sections were annotated by a single person and therefore the evaluation does not factor in the subjectivity between different annotators. Second, the section divisions in different forms are much different from the studied forms, which might influence the structure similarity. For example, many vocal compositions of OTMM with terennüms (repeated words with or without meaning such as “dost,” “aman,” “ey”) are expected to have a lower optimal similarity threshold in the lyrics relationship computation step. Moreover the poetic lines might not coincide with melodic sections in many vocal compositions especially in folk music genre. Third, the threshold can be different in different granularities. For example, the phrases are much shorter than the sections as can be seen in Figure 4.4. Human annotators might perceive the intra-similarity between sections and phrases differently.

4.3.5 Summary

In this Section, a method to automatically analyze the melodic and lyrics organization of the music score of OTMM is proposed. The method is applied on the **SymbTr** collection (Section 4.5). The extracted structural information is later used in automatic score validation, score engraving and audio-score alignment tasks (Section 4.4).

In the future, other string matching or dynamic programming algorithms (Serrà et al., 2009; Şentürk, Holzapfel, & Serra, 2014) may be tested as the similarity measure using different constraints. Additionally, the optimal similarity threshold l may be selected automatically according to the melodic and lyrics characteristics of the scores. It is also desired to solidify the findings by evaluating the methodology on a bigger dataset annotated by multiple experts and cross-comparing the annotated and the automatically extracted boundaries as done by Bozkurt, Karaosmanoğlu, et al. (2014). The ultimate aim is to develop methodologies, which are able to describe the musical structure of many music scores and audio recordings semantically and on different levels.

4.4 Applications

The score metadata extraction (Section 4.1) and structural analysis (Section 4.3) are implemented within a repository called *symbtrdataextractor* (Section 4.4.1).[13] Additionally, we created several repositories for converting the score format (Section 4.4.2) and for automatically validating/correcting the music scores (Section 4.4.3). The automatically extracted sections are later used

For the sake of modularity, these functionalities in different repositories are all integrated to a single symbolic analysis package in **T**urkish-**O**ttoman **M**akam (M)usic **A**nalysis **TO**olbox (`tomato`).[14] Figure 4.6 shows the block diagram of the functionalities in the symbolic analysis package. For more details on the implementations, please refer to Appendix C.

[13] `https://github.com/sertansenturk/symbtrdataextractor/`

[14] `https://github.com/sertansenturk/tomato/tree/v0.9.1/tomato/symbolic`

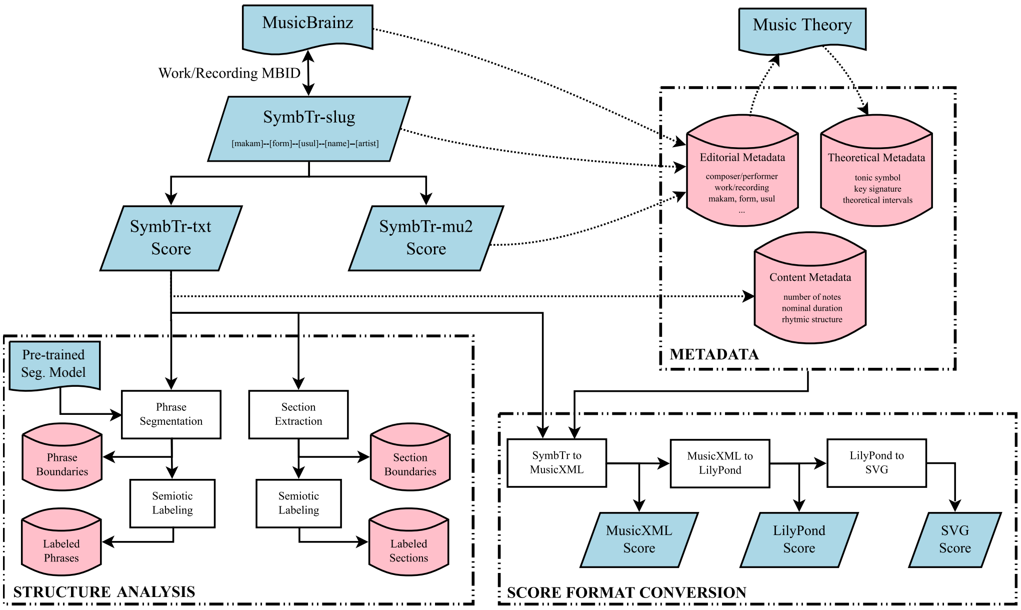


Figure 4.6: The block diagram of the score analysis and processing tasks integrated to `tomato`.

4.4.1 Automatic Content Description

The *symbtrdataextractor* repository is able to parse the headers of the **SymbTr**-mu2 files, process the contents of the **SymbTr**-txt files and crawl the relevant metadata from MusicBrainz. To crawl MusicBrainz, a wrapper around the open-source *Musicbrainz NGS bindings*[15] is created, which is specialized to fetch the OTMM related metadata.[16] The *symbtrdataextractor* package also reports any errors or inconsistencies in the obtained automatic description.

4.4.2 Music Score Format Conversion

We have developed tools in Python to convert the **SymbTr**-txt scores to the MusicXML format[17] and then to the LilyPond format[18] to improve the accesibility of the collection from popular

[15] `https://github.com/alastair/python-musicbrainzngs`
[16] `https://github.com/sertansenturk/makammusicbrainz`
[17] `https://github.com/burakuyar/MusicXMLConverter`
[18] `https://github.com/hsercanatli/makam-musicxml2lilypond`

music notation and engraving software.[19] The converters use the information obtained from *symbtrdataextractor* to add the metadata and the section names into the converted scores. The row index of each note (and rest) in the **SymbTr**-txt files are stored in the converted files as inline comments.

The scores in the LilyPond format are later converted to SVG using the LilyPond software itself. The software divides the score into multiple files during the conversion. As a postprocessing step, these files are joined together. Similar to the previous steps in the score conversion, the note indices are also stored. These indices are used later to visualize the audio-score alignment results (Section 6.8) synchronous to the audio playback in Dunya (Section 7.1.2).

The score format conversion chain is automated in `tomato`.[20] The files in the MusicXML format are hosted in the **SymbTr** GitHub repository and updated with each release.[21] The scores in the latest release is also hosted in Dunya-makam along with the score file in the SVG format.

4.4.3 Automatic Music Score Validation

The **SymbTr** collection contains unit tests,[22] which are automatically run after each new commit is pushed to the remote repository.[23] The unit tests check many possible issues such as file encoding mismatches, file name (i.e. **SymbTr**-slug, see Section 4.1) inconsistencies, missing and/or incomplete annotations (e.g. usul changes), metadata mismatches between different information sources (Section 4.1), erroneous (i.e. non-zero) grace note durations, and shifts between the note durations and the timestamps.

[19] MusicXMLConverter and makam-musicxml2lilypond packages are mainly realized by Burak Uyar and Hasan Sercan Atlı, respectively, within their masters research under the advisorship of Barış Bozkurt. My contributions to these packages are mostly related to bug fixing, refactoring, documentation and deployment. As of August 2016, the number of lines manipulated by me is approximately the same with the main developers in both packages.

[20] `https://github.com/sertansenturk/tomato/blob/v0.9.1/tomato/symbolic/scoreconverter.py`

[21] `https://github.com/MTG/SymbTr/tree/v2.4.3/MusicXMLL`

[22] `https://github.com/MTG/SymbTr/tree/v2.4.3/unittests`

[23] `https://travis-ci.org/MTG/SymbTr`

To maintain the **SymbTr** collection, a complementary repository called *SymbTr-extras*[24] is created. *SymbTr-extras* automatically fixes most of the issues reported in the unit tests. Moreover, the repository facilitates **SymbTr**-slug renaming and the **SymbTr**-txt score to MusicXML conversion.

4.4.4 Audio-Score Alignment

In the performances of OTMM compositions, the musicians occasionally insert, repeat and omit sections. Moreover they may introduce musical passages, which are not related to the composition (e.g. improvisations). In Şentürk, Holzapfel, and Serra (2014), a section-level audio-score alignment methodology is proposed, which considers such structural differences (the method is explained in Section 6.7). In the original methodology the sections in the score are manually annotated with respect to the melodic structure. Next, the candidate time intervals in the audio recording are found for each section using partial subsequence alignment. The manual section annotation step is replaced with the automatic section analysis part of the aligment method, where the melody labels are used to align relevant audio recordings and music scores (Section 6.12). I have additionally conducted experiments using the melodic relations of the extracted phrases. The preliminary results suggest that phrase-level alignment may provide better results than section-level alignment.

4.5 Automatic Description of the SymbTr Collection

Using the optimal similarity threshold ($l = 0.7$), structural analysis (Section 4.3) is applied on the **SymbTr** release, v2.4.2 (Figure 4.7). 49231 phrases are extracted and labeled from 1344 scores, which have both their makam and usul covered in the phrase segmentation training model. In parallel, 21578 sections are extracted from 1772

[24] https://github.com/MTG/SymbTr-extras

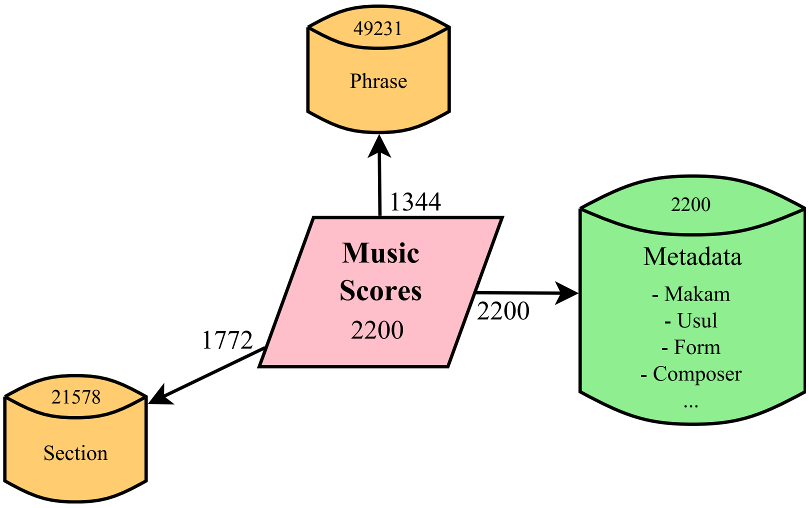


Figure 4.7: An overview of the automatic description of the **SymbTr** collection v2.4.2. The numbers in the boxes and the numbers next to the arrows indicate the total number of the entity and the number scores for which the relevant entity is extracted, respectively.

scores.[25] The data can be further used to study the structure of musical forms of OTMM.

In addition to the structural analysis, the metadata is extracted for each score (Section 4.1). Please refer to Section 3.1.2 for the statistics.

4.6 Conclusion

In this Chapter, the automatic description process applied to music scores is presented. The description consists of:

- Curated metadata related to the music scores (such as makam, usul, form, tempo and composer/performers). The metadata is fetched by parsing the **SymbTr**-txt and **SymbTr**-mu2 scores and crawling the relevant linked data in MusicBrainz

[25] The data is available at `https://github.com/sertansenturk/turkish_makam_corpus_stats/tree/66248231e4835138379ddeac970eabf7dad2c7f8/data/SymbTrData`

(Section 4.1). The metadata from different sources are also validated against each other. The procedure provides high quality and linked data related to the music scores without a need of sophisticated information retrieval techniques.

- Structure information obtained using a novel structural analysis methodology (Section 4.3). The method extracts the sections using implicit section information annotated in the the music scores. In parallel, it automatically segments the score into phrases according to makam and usul-specific models trained from the manual annotations of experts. Next, the melodic and lyrics relations between each phrase and each sections are labeled semiotically by a graph-based procedure.

The automatic description is used to enhance the **SymbTr** collection and also to find inconsistencies in the data (Sections 4.4.1 and 4.4.3). The automatic description of the **SymbTr** collection includes the metadata related to all 2200 **SymbTr**-scores, 49231 phrases in 1344 scores and 21578 sections in 1772 scores. The labeled sections are later used to automate the section-level audio-score alignment method (Section 4.4.4). Recently, the semiotic labeling is also applied to manually annotated score fragment boundaries to automatically label the fragments in the lyrics-to-audio a-lignment method proposed in (Dzhambazov et al., 2016).

In addition to the automatic description, we created automatic score format converters to MusicXML, LilyPond and SVG in order to improve the accesibility of the **SymbTr**-scores via various applications (Section 4.4.2). The MusicXML files are now included in the **SymbTr** collection and we plan to make this the main format in the next major release (**SymbTr** v3.0). The SVG scores are used to visualize the audio-score alignment results in the note, measure and section granularities (explained in Section 7.1.2).

Chapter 5

Audio Analysis

The audio recordings can provide information about the characteristics (e.g. in terms of dynamics or timing) of an interpretation of a particular piece. By analyzing large collections of audio recordings, we can obtain generalizable information about the performance aspect of the studied music. By utilizing automatic analysis methodologies, not only the time and effort to obtain such a description can be greatly reduced but we can also obtain reliable information, some of which would be too difficult or tedious for human annotators.

In the last decade, numerous sofware libraries for audio analysis have emerged such as *Marsyas* (Tzanetakis & Lemstrom, 2007), *jAudio* (McKay, 2010), *MIRtoolbox* (Lartillot, Toiviainen, & Eerola, 2008), Essentia (Bogdanov et al., 2013), *Tarsos DSP* (Six, Cornelis, & Leman, 2014), *librosa* (McFee et al., 2015), and *madmom* (Böck, Korzeniowski, Schlüter, Krebs, & Widmer, 2016). These tools have been facilitating the development and advancement of many methodologies for music information research. However, these tools do not typically incorporate the implementations of the state-of-the-art methodologies for the analysis of audio recordings of OTMM. Moreover, many suitable algorithms are constrained for Eurogenetic musics (e.g. 12-TET).

Considering the need of computational tools specialized to process and analyse OTMM, Bozkurt (2011) has developed Makam Toolbox in MATLAB. This toolbox incorporates several fundamental tasks to analyse audio recordings of OTMM such as predominant melody extraction (Bozkurt, 2008), tonic identification (Gedik & Bozkurt, 2010), tuning analysis (Bozkurt, 2012) and automatic transcription (Gedik, 2012). In addition, Makam Toolbox provides a visual interface, in which the user can select the analysis algorithms to be applied and observe the outputs. However, this toolbox is not publicly available, and it is not straightforward to setup and distribute in different platforms due to certain limitations of its design in the MATLAB environment. For these reasons, Atıcı, Bozkurt, and Şentürk (2015) have developed an open reimplementation of the toolbox in Java.[1] The reimplementation is called MakamBox.[2] MakamBox is capable of performing most of the analysis in the original toolbox[3] with a more responsive visual interface. The interaction in both toolboxes are mainly designed around the visual interface and hence it is centered around processing a single audio recording.[4] Therefore, they are not suitable for analysing large-scale audio corpora automatically.

In this Chapter, I give an overview of the current state-of-the-art audio analysis methodologies proposed for OTMM. I present numerous generalization and improvements of the existing methods. I also propose several adaptations from methods already applied to other similar music cultures. Another aim is to provide open implementations of these methodologies to support open and reproducible research. The main contributions may be summarized as:

- An adaptation of a state-of-the-art predominant melody extraction method.
- A generalization of pitch-distribution based mode recognition and tonic identification methodologies previously applied to OTMM and Hindustāni music.
- Open implementations of all audio analysis methodologies I have developed, which are described throughout this Chap-

[1] The toolbox is designed, developed and implemented by Bilge Miraç Atıcı. He is supervised by Barış Bozkurt. I have assisted Miraç Atıcı in the preparation of this publication and do not consider this work as part of my PhD research.

[2] https://miracatici.com/makambox

[3] For example, MakamBox does not incorporate automatic transcription and uses a simpler, predominant melody extraction algorithm.

[4] Makam Toolbox is capable of processing several recordings to obtain an overall tuning.

ter. These implementations cover most of the computational tasks applied on audio recordings of OTMM discussed in (Bozkurt, Ayangil, & Holzapfel, 2014).

- Integration of the aforementioned analysis methodologies into a single workflow to facilitate the automatic description of large-scale audio corpora of OTMM.

The rest of the Chapter is structured as: Given an audio fragment, the structured metadata is fetched from MusicBrainz (Section 5.1). In parallel, the predominant melody is extracted from the raw audio (Section 5.2). Using the predominant melody, the pitch distribution and pitch-class distribution (Section 5.5), stable pitches and pitch-classes (Section 5.6), makam (Section 5.7), tonic (Section 5.7), transposition (Section 5.8), tuning (Section 5.9), and melodic progression (Section 5.10) of the audio recording are extracted. Section 5.11 explains the integration of these methodologies into `tomato`. Section 5.12 presents the resulting automatic description of the CompMusic OTMM audio collection. Section 5.13 provides a brief conclusion.

Throughout the Chapter, a feature extracted from an audio fragment (a) would normally have the superscript (a). This superscript is omitted[5] for the sake of simplicity except the cases in which multiple audio recordings are processed (e.g. Section 5.7.2).

5.1 Metadata

Given an audio recording, basic information about the file such as the sampling frequency, bit rate and duration are fetched using *eyeD3*. [6] The recording MBID is also read from the metadata contained within the file (i.e. from the ID3 metadata in MP3 files). Next MusicBrainz is queried to obtain relevant metadata. Table 5.1 makes a summary of the extracted metadata. To crawl MusicBrainz, a wrapper around the open-source *MusicBrainz NGS bindings*[7] is created, which is specialized to fetch the OTMM related metadata in a structured manner.[8]

[5] e.g. the predominant melody of (a), $\boldsymbol{\varrho}^{(a)}$, is printed as $\boldsymbol{\varrho}$.

[6] `http://eyed3.nicfit.net/`

[7] `https://github.com/alastair/python-musicbrainzngs`

[8] `https://github.com/sertansenturk/makammusicbrainz`

Table 5.1: Summary of the metadata obtained for an audio recording.

Source	Key	Explanation
Audio file	path	Location of the audio file
	duration	Duration of the audio file (in seconds)
	bit_rate	Bit rate of the audio file
	sampling_frequency	Sampling frequency of the audio file
	mbid	MusicBrainz identifier
	url	URL in MusicBrainz (basically, http://musicbrainz.org/recording/[mbid])
MusicBrainz	title	Title of the recording in MusicBrainz
	releases	List of releases, which the recording is part of
	works	List of works performed in the recording
	artist-credits	List of main credited artist(s)
	artists	List of artists, who took part in realizing the recording; vocalists, intrument performers, conductors, recording engineers, etc.
	makam	List of makams associated with the recording. The information is fetched both from the related works and also the tags in MusicBrainz associated with the recording ("makam: [makam_name]")
	form	List of forms associated with the recording. The information is fetched both from the related works and also the tags in MusicBrainz associated with the recording ("form: [form_name]")
	usul	List of usuls associated with the recording. The information is fetched both from the related works and also the tags in MusicBrainz associated with the recording ("usul: [usul_name]")

5.2 Predominant Melody

In analyzing the tonal characteristics of music traditions involving harmony, features capturing harmonic content such as chroma features (Müller, Ewert, & Kreuzer, 2009; Gómez, 2006) are typically used. Chroma features are the state of the art features used in structure analysis of Eurogenetic musics (Paulus et al., 2010) and also in relevant tasks such as version identification (Serrà et al., 2009) and audio-score alignment (Thomas et al., 2012). On the other hand, predominant melody is a more representative, musically meaningful and interpretable feature for analyzing melody-dominant musics (Chordia & Şentürk, 2013; Bozkurt, Ayangil, & Holzapfel, 2014; Şentürk, Holzapfel, & Serra, 2014; Koduri, Ishwar, Serrà, & Serra, 2014).

Previously, The fundamental pitch extraction method proposed by (De Cheveigné & Kawahara, 2002) (YIN) was used by Bozkurt (2008) to analyze OTMM recordings. YIN has been shown to be highly reliable to estimate the fundamental frequency over time in monophonic music. However, YIN (and melody extraction algorithms in general) introduce octave errors and spurious estimations, when heterophonic or noisy recordings are analyzed. To overcome these problems, (Bozkurt, 2008) has proposed a post-filtering method to eliminate contours with a short duration, remove pitch estimations with low confidence and correct octave errors by shifting octave of the contours closer to the neighboring contours.

Recently, Şimşek, Bozkurt, and Akan (2016) have proposed a method based on variational mode decomposition (VMD). The predominant melody extraction method proposed by (Şimşek et al., 2016) (SIM-VMD) is reported to output comparable results to YIN and our own adaptation of MELODIA, which will be explained more in Section 5.2.1.

The predominant melody extracted from an audio fragment (a) is denoted as $\boldsymbol{\varrho} := \left[\rho_1 \ldots \rho_{|\boldsymbol{\varrho}|}\right]$, where $\rho_i \in \boldsymbol{\varrho}$ is a pitch sample and $i \in [1 : |\boldsymbol{\varrho}|]$, where $|\boldsymbol{\varrho}|$ is the length of the predominant melody.

5.2.1 Adaptations of the Existing State-of-the-Art

Since predominant melody is the primary feature extracted from the audio recordings, its quality greatly determines the quality of the subsequent analysis steps. However, developing a novel methodology would require an exhaustive effort and this task has not been the main focus of this thesis. Instead, I and Hasan Sercan Atlı have worked on improving the predominant melody extraction by adapting state-of-the-art methodologies to OTMM recordings. Except the evaluation in (Şimşek et al., 2016) (which is addressed later in this Section), these adaptations are not evaluated quantitatively due to lack of ground-truth annotations, but evaluated qualitatively by comparing the output predominant melodies visually and also by audio synthesis. Some of the algorithms are also evaluated extrinsically by comparing the results obtained from audio-score alignment methods (Section 6.7).

YIN

In the preliminary experiments on audio score alignment (Şentürk et al., 2012) (explained in Appendix A), we use YIN to extract the predominant melody of the audio recordings. We used the mex implementation[9] of YIN included in the Makam Toolbox. The hop size is taken as 10ms. Next, Makam Toolbox applies the post-processing explained in (Bozkurt, 2008). The toolbox has an additional option to quantize the pitch values in the predominant melody. The advantage of the quantized predominant melody is that it takes out minor pitch variations such as vibratos. Afterwards, a median filter with a window length of 41 frames (410ms) is applied to fix spurious jumps in the predominant melody. The predominant melody extraction method in (Şentürk et al., 2012) is named as BOZ-YIN$_f$.

The results of the preliminary experiments of section linking (Appendix A) shows that as the instrumentation of a recording gets more complex (i.e. the tendency of observing more heterophonic interactions and expressive elements in an audio recording increases), the section linking performance decreases almost monotonically. This suggests that an improvement in the extraction of audio pitch contour is necessary. Through inspecting errors in the audio recording level, it is seen that the current bottleneck of the system is the pitch estimation. Since YIN is designed for monophonic sounds, many confusions arise in the predominant melody extraction due to the heterophonic nature of OTMM, especially in ensemble performances. Moreover, YIN is found to lose its robustness, where there are substantial usage of expressive elements such as legatos, slides and tremolos.

[9] http://es.mathworks.com/help/matlab/call-mex-files-1.html?s_cid=wiki_mex_1

MELODIA

Based on the observations in our preliminary section-linking experiments using YIN, we started optimizing MELODIA, a predominant melody extraction method proposed for polyphonic music signals by (Salamon & Gómez, 2012), in our following experiments on section-level audio-score alignment (Section 6.7.4). We use the Essentia implementation of the algorithm (Bogdanov et al., 2013) throughout the thesis. The predominant melody extraction procedure explained in (Şentürk, Holzapfel, & Serra, 2014) is named as SEN-MEL.

The methodology proposed by Salamon and Gómez (2012) assumes that there is no predominant melody in time intervals where the peaks of the pitch saliences are below a certain magnitude with respect to the mean of all the peaks. Moreover, it eliminates pitch contours, which are considered as belonging to the accompaniment. However, as OTMM is heterophonic (Section 2.1), unvoiced intervals are very rare. Applied on OTMM recordings, the contour selection step in the methodology (Salamon & Gómez, 2012, Section D) treats a substantial amount of melody candidates as non-salient (due to the embellishments and wide dynamic range), and dismisses a significant portion of pitch contours as unvoiced. Instead, we include all the non-salient candidates to guess predominant melody and obtain the audio predominant melody $\boldsymbol{\varrho}$.

Next the predominant melody is downsampled from MELODIA's default frame rate of ~ 344.5 frames per second (hop size of 128 samples) to ~ 21.5 frames per second or a period of ~ 46 ms, which is sufficient to track the note changes. In our alignment experiments, melody extraction is performed using various pitch resolutions (Section 6.7.4).

We also compare the extracted predominant melody and HPCPs as input features for the audio-score alignment method. The results (explained in Section 6.7.5) show that SEN-MEL performs better compared to both BOZ-YIN$_f$ and HPCPs. Moreover, a pitch precision of 50 cents is adequate for section-level audio-score alignment. Nevertheless, the pitch precision can be increased further to use the extracted predominant melody later, in more precision-demanding computational tasks such as tonic identification (Section 5.7), tuning analysis (Section 5.9) and note-level audio-score alignment (Section 6.8). Therefore, we select the bin resolution as 1 cents instead of the default (10 cents) in the computation of the pitch salience function.[10]

Additional adaptations on MELODIA

The predominant melody computed by `SEN-MEL` still produces substantial amount of errors, especially when the music is played softer than the rest of the audio. This becomes a noticeable problem in the end of the melodic phrases, where musicians choose to play softer. For this reason, we decided to optimize the methodology of (Salamon & Gómez, 2012) step by step (Atlı et al., 2014).[11] This method is termed as `ATL-MEL` throughout the text.

Figure 5.1 shows the steps followed to compute the predominant melody. All the steps shown (inside boxes) in the Figure are named the same with the respective Essentia class,[12] except the “PitchContours Selection” step modified by us (explained below). In the “FrameGenerator” step, the audio is divided into frames and these frames are processed in parallel between the “Windowing” and “PitchSalienceFunctionPeaks” steps. We refer the reader to the Essentia documentation[13] and (Salamon & Gómez, 2012) for further information on the algorithms and their default parameters.[14]

In the computation of pitch contours,[15] we experimented on different values of the peak distribution threshold parameter to get a satisfactory pitch contour length. We emprically set the “peakDis-

[10] `http://essentia.upf.edu/documentation/reference/streaming_PitchSalienceFunction.html`

[11] This work was mainly conducted by Hasan Sercan Atlı under my guidance during his Erasmus+ stay in Music Technology Group, UPF in Summer 2014. Apart from proposing methodology, I have also contributed to the code with bug fixing, refactoring, documentation and deployment. Andrés Ferraro has also contributed to the code deployment and refactoring.

[12] in $v2.1\_beta3$ as of November 2016: `https://github.com/MTG/essentia/releases/tag/v2.1_beta3`

[13] `http://essentia.upf.edu/documentation/algorithms_reference.html`

[14] The implementation is openly avaliable at `https://github.com/sertansenturk/predominantmelodymakam`

[15] `http://essentia.upf.edu/documentation/reference/streaming_PitchContours.html`

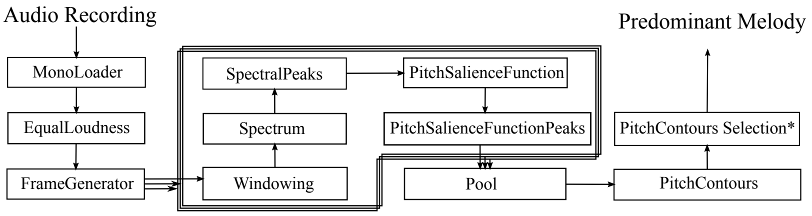


Figure 5.1: The block diagram of ATL-MEL.

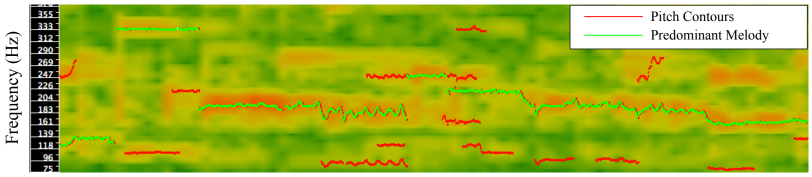


Figure 5.2: Pitch contours and the resultant predominant melody extracted from an audio fragment using ATL-MEL, plotted on top of the spectrogram of the fragment. Sonic Visualizer is used to compute the spectrogram and to display the features.

tributionTreshold" parameter as 1.4. This setting provides longer pitch contours compared to the default value of 0.9. Originally, the contour selection step in the method is trained using Eurogenetic musics (Salamon & Gómez, 2012), which is the main reason for the erroneous unvoiced estimations (explained in SEN-MEL). However, lacking the ground truth for training during the time of development, we decided to replace the contour selection step with a simple and generic heuristics. Once the pitch contours are obtained, we order the pitch contours according to their length and start with selecting the longest one. Then, we remove all portions of pitch contours which overlap with the selected pitch contour. We carry the same process for the next longest pitch contour, and so forth. By repeating the process for all pitch contours, we obtain the predominant melody of the audio recording (Figure 5.2).

Due to lack of ground truth in the time of implementation, we made a qualitative comparison between the predominant melodies obtained from SEN-MEL and ATL-MEL by synthesizing and listening the resultant predominant melodies synchronous to the audio playback. We observed that ATL-MEL is able to capture the melody better when the music is played softer in the expense of obtaining more spurious pitch estimations and octave errors.

To get rid of the spurious estimations and octave errors, we re-introduced the post-processing method proposed by Bozkurt (2008). We first tried the *PitchFilter* function[16] in Essentia, which is an open implementation of this method. However, this implementation was not good enough in removing spurious estimations, which affected the tonic identification accuracy using the last note detection method Section 5.7.2. Therefore, we made our own open source implementation.[17] This filtered variant of the predominant melody extracted using the procedure is described in (Atlı et al., 2014), and it is abbreviated as $\texttt{ATL-MEL}_f$.[18]

In his masters thesis, Atlı (2016) extracted the predominant melody of 18 audio recordings in the OTMM partial audio-score alignment dataset (Section 3.2.7) using $\texttt{ATL-MEL}_f$ and recorded the number of the predominant melody samples that are within 1 Hc vicinity of the annotated notes (Figure 5.3). He reports that 143308 out of 157231 samples (91.14%) coincide with the note annotations, implying $\texttt{ATL-MEL}_f$ outputs reliable predominant melody estimations. Nevertheless, these findings should not be regarded as an intrinsic evaluation of the method, since the note annotations do not include any information on the intonation deviations or the embellishments.

Şimşek et al. (2016) compared their method SIM-VMD with YIN and $\texttt{ATL-MEL}_f$ on four heterophonic OTMM recordings. They report the evaluation measures used in Music Information Retrieval EXchange (MIREX) Audio Melody Extraction task.[19] The results given in (Şimşek et al., 2016, Table 1) indicate that $\texttt{ATL-MEL}_f$ is superior to YIN, and it outputs either comparable or better results than SIM-VMD over all evaluation measures.

[16] http://essentia.upf.edu/documentation/reference/std_PitchFilter.html

[17] https://github.com/hsercanatli/pitchfilter

[18] Reimplementation of the pitch filter has been done by Hasan Sercan Atlı within his masters research under the advisorship of Barış Bozkurt. Later, I have contributed to the code with bug fixing, refactoring, documentation and deployment.

[19] http://www.music-ir.org/mirex/wiki/2016:Audio_Melody_Extraction#Evaluation_Procedures

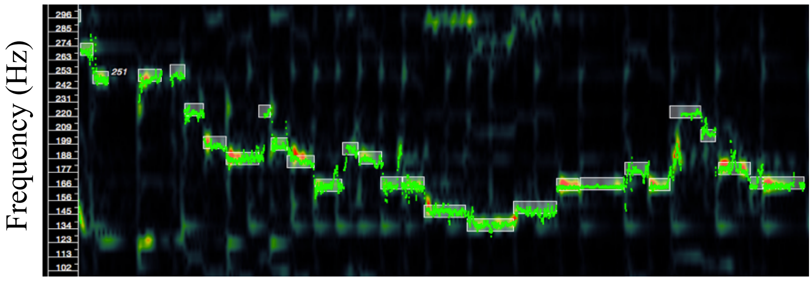


Figure 5.3: The predominant melody (in green), the note annotations (white transparent boxes) and the melodic range spectrogram (background) of an audio fragment in the OTMM partial audio-score alignment dataset. Sonic Visualizer is used to compute the melodic range spectrogram and to display the features. The Figure is reproduced from (Atlı, 2016) courtesy of Hasan Sercan Atlı.

5.3 Harmonic Pitch Class Profiles

As mentioned in the Section 5.2, HPCPs (Gómez, 2006) are compared to predominant melody in section linking experiments and predominany melody is intrinsically observed to be a more representative feature to describe the melodic characteristics of OTMM. The experiments will be introduced in Section 6.7.4.

We use the default parameters given in (Gómez, 2006) to compute the HPCPs. The hop size and the frame size are chosen to be 2048 (e.g. ~ 21.5 frames per second) and 4096 samples respectively. The first bin of the HPCPs is assigned to the tonic pitch or pitch-class κ identified automatically using distribution matching (Section 5.7.2) or score-informed tonic identification (Section 6.4) and HPCPs, $\hat{\mathbf{\Gamma}}^{(a)} = \left[\hat{\gamma}_1^{(a)}, \ldots, \hat{\gamma}^{(a)}_{|\hat{\mathbf{\Gamma}}^{(a)}|}\right]$ are obtained, where $|\hat{\mathbf{\Gamma}}^{(a)}|$ is the number of frames in time. Each frame consists of a constant number of bins. The number of bins denoted as n_{HPCP} determines the pitch resolution of the HPCPs, i.e. the width of each bin equals to $1200/n_{\text{HPCP}}$. Figure 5.4 shows the tonic normalized HPCPs extracted from a short audio fragment.[20]

[20] http://musicbrainz.org/work/9aaf5c0b-4642-40fd-97ba-c861265872ce

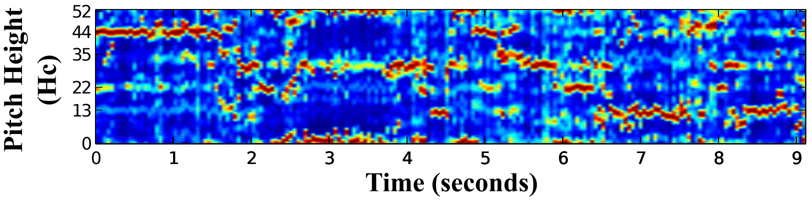


Figure 5.4: HPCPs extracted for a short audio fragment. The first bin of the HPCPs is centered at its annotated tonic frequency.

5.4 Pitch Intervals

In order to process the pitch content independent of the absolute frequency (and as a result, independent of the octave), the pitch values should be converted to intervals. This is achieved by converting the pitch values in Hertz unit to cents. A pitch value ρ in Hertz is converted to the cent scale by taking a frequency value f as the reference in the equation below:

$$\hat{\rho}^f := 1200 \log(\rho/f) \tag{5.1}$$

Hence, the cent value shows the ratio between two frequencies. This operation can be applied to any feature representing the pitch content in Hertz such as the predominant melody (Section 5.2) and the bins of the pitch(-class) distributions (Section 5.5), and the stable pitches (Section 5.6). For example, given the predominant melody $\boldsymbol{\varrho}$ extracted from an audio fragment (a), $\hat{\boldsymbol{\varrho}}^f := \left[\hat{\rho}_1^f \ldots \hat{\rho}_{|\hat{\boldsymbol{\varrho}}^f|}^f\right]$ denotes the predominant melody converted to cent scale with respect to the frequency value f. Here $\hat{\rho}_i^f \in \hat{\boldsymbol{\varrho}}^f$ is a pitch sample in cents and $i \in \left[1 : |\hat{\boldsymbol{\varrho}}^f|\right]$, where $|\hat{\boldsymbol{\varrho}}^f| = |\boldsymbol{\varrho}|$ is the length of the predominant melody.

Given a pitch value ρ, the cent distance of its pitch class to the reference frequency f (also termed as "octave-wrapped" cent distance throughout the text) is computed as:

$$\triangle(\rho, f) := \hat{\rho}^f \mod 1200 \tag{5.2}$$

where mod is the modulo operation and $\hat{\rho}^f$ is the cent-distance of ρ with respect to f (Equation 5.1). Notice that $\triangle(f, \rho) = - \triangle(\rho, f)$ mod $1200 = 1200 - \triangle(\rho, f)$.

The shortest “octave-wrapped” cent-distance $\blacktriangle(\rho, f)$ between ρ and f is computed as:

$$\blacktriangle(\rho, f) := min\big(\triangle(\rho, f), \triangle(f, \rho)\big) \tag{5.3}$$

5.5 Pitch and Pitch-Class Distributions

PDs and pitch-class distributions (PCDs) show the relative occurrence of the pitch and pitch class values with respect to each other, respectively. PDs and PCDs are commonly used for analysis of tonic and pitch organisation. Krumhansl and Shepard (1979) used 12-dimensional PCDs to study the tonal organisation of euro-genetic musics. PCDs are also used for relevant tasks such as key detection, chord recognition (Gómez, 2006; Temperley & Marvin, 2008) and genre classification (Tzanetakis, Ermolinskyi, & Cook, 2003) for Eurogenetic musics.

For musical styles involving microtonality, the pitch space must be extended beyond 12-dimensions to model, analyze and predict the melodic properties of the studied music (Bozkurt et al., 2009; Gedik & Bozkurt, 2010; Şentürk, 2011; Şentürk, Holzapfel, & Serra, 2014). Pitch distributions have been used in many tasks applied to OTMM such as tonic identification (Bozkurt, 2008; Atlı et al., 2015), makam recognition (Gedik & Bozkurt, 2010) and tuning analysis (Bozkurt et al., 2009). Likewise, these features will be used extensively throughout the thesis in stable pitch and pitch-class extraction (Section 5.6), tonic identification (Sections 5.7 and 6.4), tempo estimation (Section 6.5), tuning analysis (Section 5.9), audio melodic progression analysis (Section 5.10) and note modeling (Section 6.11).

These distributions can be computed from any time-series representation of pitch such as spectrograms, chroma features and predominant melody. Since Ottoman-Turkish makam music is a melody-dominant tradition (Section 2.1), we use the predominant melody to compute these distributions. First, the predominant melody $\boldsymbol{\varrho}$ is converted from Hz-scale to cent scale by taking a frequency value $*$ as the reference, and $\hat{\boldsymbol{\varrho}}^{*}$ is obtained. $*$ equals to the tonic pitch κ, if available; otherwise $*$ is arbitrarily assigned to 440 Hz. The conversion to cents allows to compute the distribution with constant bin width in the cents scale. The values in both distributions are computed as:

$$\hat{h}_n^* := \frac{\sum_{i=1}^{|\hat{\boldsymbol{\varrho}}^*|} \ell_n\left(\hat{\rho}^*\right)}{\left|\hat{\boldsymbol{\varrho}}_i^*\right|} \tag{5.4}$$

where $\hat{h}_n^*$ is the relative occurence computed for the n^{th} bin of the distribution $\hat{\boldsymbol{H}}^*$, computed from the samples $\hat{\rho}^* \in \hat{\boldsymbol{\varrho}}^*$. As a convention, 0^{th} bin is centered around the reference frequency; or in other words, the reference pitch is mapped to 0 cents.

The accumulator function $\ell_{P,n}$ for PDs is defined as:

$$\ell_{P,n}(\hat{\rho}) := \begin{cases} 1, & c_n \leq \hat{\rho} \leq c_{n+1} \\ 0, & \text{otherwise} \end{cases} \tag{5.5}$$

where $\hat{\rho}$ is a pitch sample in cents and (c_n, c_{n+1}) are the boundaries of the n-th bin. Similarly the accumular function for PCDs is defined as:

$$\ell_{PC,n}(\hat{\rho}) := \begin{cases} 1, & c_n \leq (\hat{\rho} \mod 1200) \leq c_{n+1} \\ 0, & \text{otherwise} \end{cases} \tag{5.6}$$

Note that the PCD is a “circular” feature, e.g. the first and the last bins are adjacent to each other. On the other hand, PD would span to multiple octaves. Therefore a PD would typically have bins with negative indices, which represent frequencies below the reference frequency. Also notice that both PD and PCD are normalized such that the resultant distribution can be treated as a probability density function (Equation 5.4).

The bin size $b\left(\hat{\boldsymbol{H}}\right)$ of a distribution $\hat{\boldsymbol{H}}$ determines how precise the distribution is (to the extend allowed by the cent-precision of the predominant melody) in representing the pitch space, the tuning of the stable pitches and the microtonal characteristics in a lower-level. The computed distributions might need to have a small bin size, e.g. less than a quarter tone (50 cents) for many music cultures (Gedik & Bozkurt, 2010; Chordia & Şentürk, 2013). We select a constant bin size for the computed distributions, i.e. $b\left(\hat{\boldsymbol{H}}\right) = c_{n+1} - c_n, \forall n$. The bin centers of both PDs and PCDs are selected such that the reference frequency r is represented as a bin centered around 0 cents. We denote the number of bins in a distribution $\hat{h}$ as $|\hat{\boldsymbol{H}}|$. Note that $|\hat{\boldsymbol{H}}_{PC}|$ equals to $\lfloor 1200/\ b\left(\hat{\boldsymbol{H}}\right)\rfloor$ in a PCD.

To remove the spurious peaks in the distribution, the distribution is convolved with a Gaussian kernel and a "smoothed" distribution is obtained (Chordia & Şentürk, 2013). The standard deviation of the Gaussian kernel, termed as the kernel width $\sigma\left(\hat{\boldsymbol{H}}\right)$, determines how smooth the resulting distribution will get. The kernel width should be comparable to the bin size $b\left(\hat{\boldsymbol{H}}\right)$ since a value lower than one third of the bin size would not contribute much to smoothing[21] and a high value would "blur" the distribution too much. Moreover, this parameter has a direct impact on the number and the location of peaks in distribution, which are later used in tonic identification (Section 5.7) and tuning analysis (Section 5.9). Finally, bin values are converted from cents back to Hz using the inverse of the Equation 5.1 and $\boldsymbol{H}$ is obtained (Figure 5.5).

Unless stated otherwise, the default value of the bin size $b\left(\hat{\boldsymbol{H}}\right)$ is selected as 7.5 cents $\approx 1/3$ Hc, which is reported as an empirical optimal for this feature to capture tuning differences (Bozkurt, 2008). The standard deviation of the Gaussian kernel $\sigma\left(\hat{\boldsymbol{H}}\right)$ is selected as 7.5 cents such that it practically affects an area of six standard deviations (22.5 cents ≈ 2 Hc peak to tail) and does not mask quarter tone intervals (50 cents) (Şentürk et al., 2013). In our implementation, we select the overall width of the Gaussian kernel as 37.5 cents from peak to tail (i.e the kernel has 11 samples) for performance reasons.[22]

5.6 Stable Pitches and Pitch-Classes

To extract the performed notes and their intervallic relations, the first step is to extract the stable pitches and the stable pitch-classes

[21] The values of the bins in a Gaussian kernel, which are more than three standard deviations away from the mean are greatly diminished.

[22] The implementation of pitch distribution and pitch-class distribution is available at `https://github.com/altugkarakurt/morty/blob/v1.2.1/morty/pitchdistribution.py`.

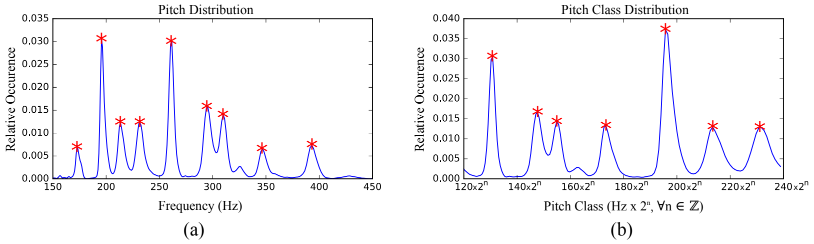


Figure 5.5: The pitch distribution and pitch-class distribution computed from a predominant melody. The stable pitch and stable pitch classes are also marked on the pitch distribution and the pitch-class distribution, respectively.

performed in the audio fragment. A simple method is to detect the peaks in the computed pitch distributions and pitch-class distributions to extract these two features, respectively.

We detect the peaks in the distribution using the Essentia implementation of the method explained in (Smith III & Serra, 1987). We only consider the peaks, which have a ratio between its height and the maxima of the distribution above a constant threshold. As will be explained in Section 5.7.6, we empirically set the ratio $\delta(\boldsymbol{H})$ to 0.15. The peaks indicate the stable pitches or stable pitch-classes performed in the fragment depending on the distribution input (Figure 5.5). We denote the set of stable pitches extracted from the pitch distribution $\boldsymbol{H}_P$ as $\boldsymbol{\Phi}_P := \{\phi_{P,1}, \phi_{P,2}, \ldots, \}$. Similarly the set of stable pitch classes extracted from the pitch class distribution $\boldsymbol{H}_{PC}$ is denoted as $\boldsymbol{\Phi}_{PC} := \{\phi_{PC,1},\ \phi_{PC,2}, \ldots, \}$.

This procedure will not be able to extract stable pitches/pitch-classes, which are performed considerably less than maxima of the distribution due to peak selection threshold or are masked by other peaks. Therefore the extracted set of pitch(-classes) are limited to "prominent" stable notes, which typically correspond to the notes in the scale of the makam.

5.7 Tonic and Makam

In many music cultures, the melodies adhere to a particular melodic framework, which specifies the melodic characteristics of the music. While the function and the understanding of these frameworks are distinct from a culture-specific perspective, in a broader sense they may be considered as the "modes" of the studied music culture. Some of the music traditions that can be considered as "modal" are Indian art musics, the makam traditions and medieval church chants (Powers, et al., n.d.). Mode recognition is an important complementary task in computational musicology, music discovery, music similarity and recommendation. In the context of Ottoman-Turkish makam music, mode is synonymous to makam.

Tonic is another important musical concept. It acts as the reference frequency for the melodic progression in a performance. In many music cultures there is no standard reference tuning frequency, which makes it crucial to identify the tonic frequency to study melodic interactions. Likewise, there is not agreed reference frequency in OTMM as stated earlier in Section 2.1. The identification of the tonic frequency is required for many tasks such as tuning analysis (Bozkurt et al., 2009), automatic transcription (Benetos & Holzapfel, 2015) and audio-score alignment (Şentürk, Holzapfel, & Serra, 2014). Estimating the tonic of a recording is the first step for various computational tasks such as tuning analysis (Bozkurt, 2012), automatic transcription (Benetos & Holzapfel, 2015) and melodic motif discovery (Gulati, Serrà, Ishwar, et al., 2016). In the context of Ottoman-Turkish makam music, tonic is synonymous to karar.

There has been a extensive interest on mode recognition in the last decade (Koduri, Gulati, Rao, & Serra, 2012). Most of these work focus on culture-specific approaches for music traditions such as Ottoman-Turkish makam music (Gedik & Bozkurt, 2010), Carnatic music (Dighe, Karnick, & Raj, 2013; Gulati, Serrà, Ishwar, et al., 2016), Hindustani music (Chordia & Rae, 2007; Chordia & Şentürk, 2013; Gulati, Serrà, Ganguli, et al., 2016) and Dastgah music (Abdoli, 2011). A considerable portion of these studies are based on comparing pitch distributions (Chordia & Rae, 2007; Chordia & Şentürk, 2013; Dighe et al., 2013; Gedik & Bozkurt, 2010), which are shown to be reliable in the mode recognition task. There also exists recent approaches that are based on characteristic melodic motif mining using network analysis (Gulati, Serrà, Ishwar, et al., 2016), aggregating note models using automatic transcription (Koduri et al., 2014), audio-score alignment (Şentürk, Koduri, & Serra, 2016) or neural network based classification (Suma

& Koolagudi, 2015; Shetty & Achary, 2009), all of which are designed specific to the studied music culture and are not generalizable to other music cultures without considerable effort. Similarly, several studies on tonic identification use pitch distribution based methods (Bozkurt, 2008; Chordia & Şentürk, 2013). More recently there has been an interest in culture specific methods for this task (Şentürk et al., 2013; Gulati, 2011; Atlı et al., 2015) that make use of heuristics and the musical characteristics of the studied tradition.

5.7.1 Problem Definition

Mode recognition is defined as classifying the mode $\mu^{(a)}$ of an audio fragment (a) from a discrete set of modes $\boldsymbol{\mathcal{M}} := \{\mu_1, \ldots, \mu_V\}$, where $\mu^{(a)} \in \boldsymbol{\mathcal{M}}$ and $|\boldsymbol{\mathcal{M}}|$ is the total number of modes. In mode recognition, we assume that the true tonic frequency (or pitch class) $\mathfrak{t}^{(a)}$ of the audio recording is available.

Tonic identification is defined as estimating the frequency or the pitch class (if the octave information of the tonic is not well-defined for the music culture or the performance) of the performance tonic. We denote the estimated tonic of an audio fragment as $\kappa^{(a)}$. Tonic is a continuous variable. However, in practice, the tonic is typically constrained to be one of the stable pitches or pitch classes performed in the audio fragment (Chordia & Şentürk, 2013; Gedik & Bozkurt, 2010). With this assumption, tonic identification can be reformulated as estimating the tonic frequency or the pitch class $\kappa^{(a)}$ from a finite set of stable pitches/pitch-classes $\mathbf{\Phi}^{(a)} := \left\{\phi_1^{(a)}, \ldots, \phi_{|\mathbf{\Phi}^{(a)}|}^{(a)}\right\}$ performed in an audio fragment (a), where $\kappa^{(a)} \in \mathbf{\Phi}^{(a)}$ and $|\mathbf{\Phi}^{(a)}|$ is the number of the stable pitches/pitch-classes in the audio fragment. In the context of OTMM, we focus on identifying the pitch class of the tonic, since the frequency of the tonic is ambiguos in heterophonic recordings, as explained in Section 2.1. In tonic identification, we assume that the true mode $\mathfrak{m}^{(a)}$ of the audio recording is known.

A third scenario arises when both the tonic $\kappa^{(a)}$ and the mode $\mu^{(a)}$ of the recording (a) are unknown. In this case, we identify the tonic and recognize the mode together, which we term as *joint estimation of mode and tonic*.

Note that these scenarios are actually multi-class problems since the mode and the tonic may change (and not necessarily simultaneously) throughout the performance. This is a more challenging problem, where we would not only like to obtain the set of the modes and tonics in the performance but also mark the intervals, where these musical "attributes" are observed.[23] Later in Section 6.7, audio-score alignment will be used to identify the tonic and mode with their time intervals.

5.7.2 Methodologies

For makam recognition and joint recognition tasks, we generalize two state-of-the-art methods on distribution matching (Gedik & Bozkurt, 2010; Chordia & Şentürk, 2013). The generalized method is explained in (Karakurt et al., 2016).[24]

For tonic identification, in addition to the generalized distribution matching method explained above, we also use the last-note detection method proposed in (Atlı et al., 2015) (ATL-TON).[25]

Last note detection

ATL-TON uses the musical knowledge that a makam music performance ends in the karar note (Section 2.1). The methodology first extracts the predominant melody from the audio recording using ATL-MEL$_f$ (Section 5.2). The end of the predominant melody is divided into chunks according to the pitch jumps on the predominant melody. Initially the tonic frequency is estimated as the median of all the frequencies in the last chunk. Then, a pitch distribution is computed (Section 5.5) using only the frequency values in

[23] A manually annotated example for OTMM is given in http://musicbrainz.org/recording/37dd6a6a-4c19-4a86-886a-882840d59518

[24] This work was mainly conducted by Altuğ Karakurt within his Erasmus+ internship under my guidance. I was responsible to design the methodology, dataset and experiments. I have also contributed in developing the library, bug fixing, refactoring, documentation and deployment.

[25] This work was mainly conducted by Hasan Sercan Atlı within his masters research under the advisorship of Barış Bozkurt. I have assisted Hasan Sercan Atlı in implementing the methodology and improving its performance. I have also contributed with bug fixing, refactoring, documentation and deployment.

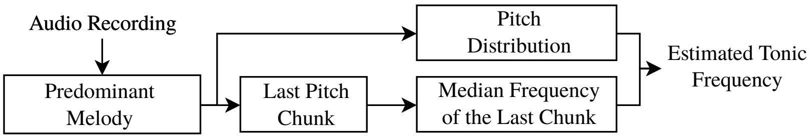


Figure 5.6: The block diagram of ATL-TON.

the predominant melody, which are close to the initial estimation (± a semitone). Then the tonic estimation is refined as the frequency of the closest peak in the histogram. The flow diagram of ATL-TON is shown in Figure 5.6.

This method provides a simple and generalizable solution without requiring neither training nor additional information such as the makam of the audio recording (as needed by the distribution matching method; Section 5.7.2) or the music score (as needed by the score-informed method; Section 6.4). Note that this method will fail for any audio fragment, which does not end with the tonic note,[26] and the accuracy of this method is susceptible to the quality of the predominant melody in the end of the fragment.[27]

Distribution matching

In (Karakurt et al., 2016), we combine and generalize the two state of the art methods, originally proposed for audio recordings of Ottoman-Turkish makam music (Bozkurt, 2008; Gedik & Bozkurt, 2010)[28] and short audio fragments of Hindustani music (Chordia & Şentürk, 2013). The generalized methods are supervised and use k nearest neighbors (kNN) estimation for classification. Our implementation is generic such that the parameters selected in the feature extraction, training and testing steps can be optimized for the properties of the studied music tradition. We also allow the user to classify either short audio fragments or complete audio recordings and switch between different features, training schemes and tasks as introduced in (Bozkurt, 2008; Gedik & Bozkurt, 2010; Chor-

[26] e.g. http://musicbrainz.org/recording/deadd528-5faf-4377-8c68-ea7145112c34

[27] The open implementation of this method is available in https://github.com/hsercanatli/tonicidentifier_makam

[28] (Bozkurt, 2008) introduces tonic identification methodology, which is later extended to makam recognition in (Gedik & Bozkurt, 2010).

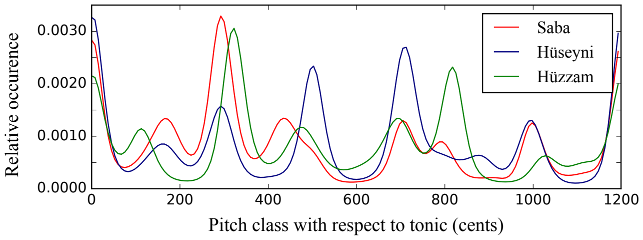


Figure 5.7: An example model with a single PCD per makam trained for three makams

dia & Şentürk, 2013). Later in Section 5.7.3, we demonstrate the experiments for the parameter selection and optimization on a test dataset of audio recordings of OTMM (Section 5.7.4) and the results of (Chordia & Şentürk, 2013) on a Hindustani and a Carnatic music dataset using the optimal parameters reported in (Chordia & Şentürk, 2013) (Appendix B.7).

In the training step, we use audio fragments with annotated makam and tonic. We first extract a predominant melody for each audio fragment. These are used to compute either pitch distribution or pitch-class distribution (Section 5.5). Next, we create makam models from these computed distributions.

Given an audio recording with an unknown makam and/or tonic, we extract its predominant melody and compute the distribution. Then, we compute a distance or dissimilarity between the distribution of the test audio and the selected distributions in the training models and compute the k nearest neighbors according to the computed measure. Finally, we estimate the unknown makam and/or tonic as the most common candidate among the k nearest neighbors.

Now we proceed to explain the generalized methodology in detail.

Training model: The generalized method is supervised and hence require training data, i.e. audio fragments with annotated makam and tonic. From a training audio fragment (x), we first extract the predominant melody $\boldsymbol{\varrho}^{(x)}$ and normalize with respect to the annotated tonic frequency $\kappa^{(x)}$ (Equation 5.1). Next, the normalized predominant melodies $\hat{\boldsymbol{\varrho}}^{\kappa,(x)}$ are grouped according to the annotated makam $\mu^{(x)}$ of each individual fragment.

The fundamental difference between the methodologies pro-

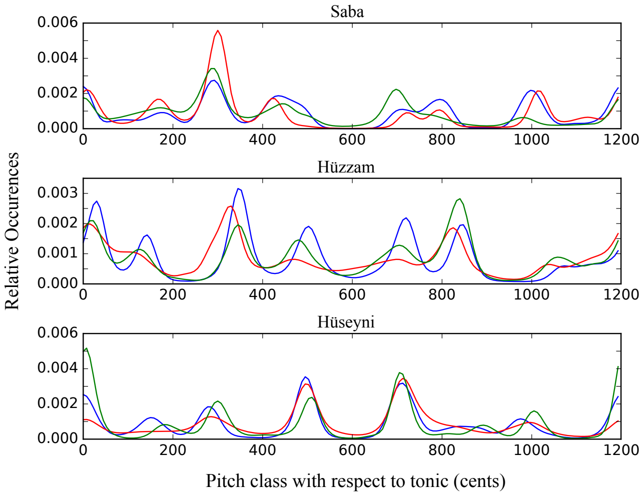


Figure 5.8: An example model with three PCDs per makam trained for three makams.

posed in (Bozkurt, 2008; Gedik & Bozkurt, 2010) and (Chordia & Şentürk, 2013) is the training model $\mathcal{T}$. The methodology proposed in (Bozkurt, 2008; Gedik & Bozkurt, 2010) joins all the normalized predominant melodies and compute a single distribution per mode. On the other hand, (Chordia & Şentürk, 2013) creates a separate distribution from each annotated audio fragment. From a machine learning perspective (Bozkurt, 2008; Gedik & Bozkurt, 2010) represents each makam with a single data point (Figure 5.7), whereas (Chordia & Şentürk, 2013) represents them with many (Figure 5.8) in an $|\boldsymbol{H}|$-dimensional space, where $|\boldsymbol{H}|$ is the number of bins in the distributions. From now on, we term the training models using the training step in (Bozkurt, 2008; Gedik & Bozkurt, 2010) and (Chordia & Şentürk, 2013) as "single distribution per mode" and "multi-distributions per mode", respectively. We denote the obtained model as $\mathcal{T} := \Big\{\langle\hat{\boldsymbol{H}}_1, \mu_1\rangle, \langle\hat{\boldsymbol{H}}_2, \mu_2\rangle, \dots\Big\}$, where $\langle\hat{\boldsymbol{H}}_j, \mu_j\rangle$ is a tuple. $\hat{\boldsymbol{H}}_j$ and μ_j denotes the trained distribution and the makam label of the j^{th} data point, respectively.[29] The model $\mathcal{T}$ consists of the distribution representations for $|\mathcal{M}|$ makams, where $|\mathcal{M}|$ is the number of unique makam labels μ_i (where

[29] The annotated tonic and the source are not used later in the classification step (Section 5.7.2). Therefore these labels are omitted from the representation in $\mathcal{T}$.

$i \in [1 : |\boldsymbol{\mathcal{M}}|]$) in the training fragments.

Nearest Neighbor Selection: In mode recognition, tonic identification and joint estimation tasks, the common step is to find the nearest neighbor(s) of a selected distribution among a set of distributions to be compared against. To find these nearest neighbors, we compute a distance or a dissimilarity between the test distribution and each distribution in the comparison set selected from the training model (Cha & Srihari, 2002). We have implemented the distance and the similarity metrics in (Gedik & Bozkurt, 2010; Chordia & Şentürk, 2013), namely, City-Block (L_1 Norm) distance, Euclidean (L_2 Norm) distance, L_3 Norm, Bhattacharyya distance, intersection and cross correlation. Note that intersection and cross correlation are similarity metrics, hence we convert them to dissimilarities (i.e. $1-$similarity) instead. The choice of the distance or dissimilarity measure plays a crucial role in the neighbor selection.

Given a distribution $\hat{\boldsymbol{H}}_j$ in the model $\boldsymbol{\mathcal{T}}$ and the distribution $\hat{\boldsymbol{H}}^{*,(a)}$ extracted from an audio fragment (a) by taking the frequency $*$ as the reference, the implemented metrics between these distributions are given below:

- City-Block (L_1 Norm) Distance:

$$\diamondsuit_{L1}\left(\hat{\boldsymbol{H}}_j, \hat{\boldsymbol{H}}^{*,(a)}\right) = \frac{1}{|\hat{\boldsymbol{H}}_j|} \sum_n |\hat{h}_{j,n} - \hat{h}_n^{*,(a)}| \tag{5.7}$$

- Euclidean (L_2 Norm) Distance:

$$\diamondsuit_{L2}\left(\hat{\boldsymbol{H}}_j, \hat{\boldsymbol{H}}^{*,(a)}\right) = \sqrt{\sum_n \left(\hat{h}_{j,n} - \hat{h}_n^{*,(a)}\right)^2} \tag{5.8}$$

- L_3 Norm:

$$\diamondsuit_{L3}\left(\hat{\boldsymbol{H}}_j, \hat{\boldsymbol{H}}^{*,(a)}\right) = \left(\sum_n \left|\hat{h}_{j,n} - \hat{h}_n^{*,(a)}\right|^3\right)^{1/3} \tag{5.9}$$

- Bhattacharyya Distance:

$$\diamondsuit_{bhat}\left(\hat{\boldsymbol{H}}_j, \hat{\boldsymbol{H}}^{*,(a)}\right) = -\log \sum_n \sqrt{\hat{h}_{j,n}\, \hat{h}_n^{*,(a)}} \tag{5.10}$$

- Inverse of Intersection:

$$\diamondsuit_{intr^{-1}}\left(\hat{\boldsymbol{H}}_j, \hat{\boldsymbol{H}}^{*,(a)}\right) = 1 - \frac{1}{|\hat{\boldsymbol{H}}_j|}\sum_n min\left(\hat{h}_{j,n}, \hat{h}_n^{*,(a)}\right) \tag{5.11}$$

- Negative of Cross-Correlation:

$$\diamondsuit_{1\text{ - }ccor}\left(\hat{\boldsymbol{H}}_j, \hat{\boldsymbol{H}}^{*,(a)}\right) = 1 - \frac{1}{|\hat{\boldsymbol{H}}_j|}\sum_n \hat{h}_{j,n}\,\hat{h}_n^{*,(a)} \tag{5.12}$$

Remember that the reference is always the 0th bin. Also remark that the PDs would not typically have the same length. During the distance or dissimilarity computation, the values of the "missing" bins in the distributions are treated as 0.

After the distances or the dissimilarities are computed, the compared distributions are ranked and the k nearest neighbors are selected. We then estimate the test sample as the most common label of the neighbors. In case of a tie between two or more groups, we select label of the group, which accumulates the lowest distance or dissimilarity. Note that if a single-distribution is computed for each mode as explained in (Gedik & Bozkurt, 2010), the k value is always 1, since each mode is only represented by one sample.

Now we proceed to explain the procedure for each task in detail.

Makam Recognition: Given an audio fragment (a) with an unknown mode, we compute the distribution $\hat{\boldsymbol{H}}^{\mathfrak{t},(a)}$ by taking the annotated tonic $\mathfrak{t}^{(a)}$ as the reference (Section 5.5). Next we compute the distance or the dissimilarity between $\hat{\boldsymbol{H}}^{\mathfrak{t},(a)}$ and the trained distribution $\hat{\boldsymbol{H}}_j$ of each tuple $\in \boldsymbol{\mathcal{T}}$, where $\boldsymbol{\mathcal{T}}$ is the trained model, and obtain the k nearest neighbors to (a). We estimate the makam$\mu^{(a)}$ of (a) as the most common makam label within the nearest neighbors.

Tonic Identification: Given an audio fragment (a) with the annotated makam $\mathfrak{m}^{(a)}$, we first extract the predominant melody $\boldsymbol{\varrho}^{(a)}$. Then we compute a distribution $\boldsymbol{H}^{(a)}$ (Section 5.5). We detect the peaks in the distribution and obtain the set of stable pitches/pitch-classes as $\boldsymbol{\Phi}^{(a)} := \left\{\phi_1^{(a)}, \dots, \phi_{|\boldsymbol{\Phi}^{(a)}|}^{(a)}\right\}$, where $|\boldsymbol{\Phi}^{(a)}|$ is the number of detected peaks (Section 5.6). Assuming each peak $\phi_i^{(a)}$ as the tonic candidate, we compute $\hat{\boldsymbol{H}}^{\phi_i^{(a)},(a)}$ such that the index of the bin representing $\phi_i^{(a)}$ becomes 0 in the shifted distribution.

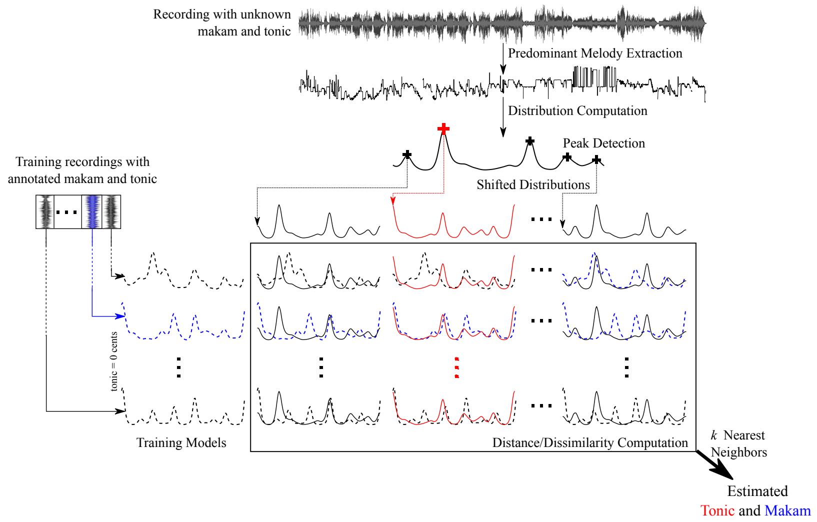

Recording with unknown makam and tonic
Predominant Melody Extraction
Distribution Computation
Peak Detection
Training recordings with annotated makam and tonic
Shifted Distributions
tonic = 0 cents
Training Models
Distance/Dissimilarity Computation
k Nearest Neighbors
Estimated Tonic and Makam

Figure 5.9: Block diagram of the joint estimation methodology. The shifted distribution in red and the training distribution in blue shows a close match.

From the training model $\boldsymbol{\mathcal{T}}$, we select all the $\langle \hat{\boldsymbol{H}}_j, \mu_j \rangle \in \boldsymbol{\mathcal{T}}$ such that the label $\mu_j = \mathfrak{m}^{(a)}$. Next we compute the distance or the dissimilarity between each shifted distribution $\hat{\boldsymbol{H}}^{\phi_i^{(a)}(a)}$ and the selected $\langle \hat{\boldsymbol{H}}_j, \mu_j \rangle$s. We obtain the k pairs with the lowest distance or dissimilarity and select the most occurring peak $\phi_i^{(a)}$ in the neighbors as the estimated tonic $\kappa^{(a)}$.

Joint Estimation of Makam and tonic: Given an audio fragment (a) with unknown makam and tonic, we compute the stable pitches/pitch-classes, $\boldsymbol{\Phi}^{(a)} := \left\{ \phi_1^{(a)}, \ldots, \phi_{|\boldsymbol{\Phi}^{(a)}|}^{(a)} \right\}$ and then the distributions $\hat{\boldsymbol{H}}^{\phi_i^{(a)},(a)}$ assuming each $\phi_i^{(a)} \in \boldsymbol{\Phi}^{(a)}$ as the tonic candidate, as explained in the tonic identification procedure explained above. Next, we compute the distance or the dissimilarity between each pair of shifted distribution $\hat{\boldsymbol{H}}^{\phi_i^{(a)},(a)}$ and the distributions $\hat{\boldsymbol{H}}_j$ in all the training samples $\langle \hat{\boldsymbol{H}}_j, \mu_j \rangle \in \boldsymbol{\mathcal{T}}$. We select the k pairs with the lowest distance or dissimilarity and estimate the most occurring $\langle$mode, tonic candidate$\rangle$ pair, i.e. $\langle \mu_i, \phi_j^{(a)} \rangle$ ($\mu_i \in \boldsymbol{\mathcal{M}}$, $\phi_j^{(a)} \in \boldsymbol{\Phi}^{(a)}$) as the makam $\mu^{(a)}$ and the tonic $\kappa^{(a)}$ of the audio fragment (a). An example procedure is shown in Figure 5.9.

5.7.3 Experiments

To evaluate the generalized methodology, we conducted exhaustive experiments on the largest makam recognition dataset of Ottoman-Turkish makam music (OTMM) (Section 5.7.4).

In most of the tonic identification and mode recognition studies in the past, the features extracted from the data,[30] the source code and the experimental results have not been generally shared. We consider the unavailability of public tools, datasets and reproducible experimentations as major obstacles for computational research on OTMM and MIR in general. For this reason, we have implemented the generalized distribution matching methodology and packaged it as a open source toolbox called **MO**de **R**ecognition and **T**onic **Y**dentification Toolbox (Karakurt et al., 2016) (`MORTY`).[31] `MORTY` is free software written in *Python* 2.7 and licensed under *Affero GPLv3*.[32] Our primary aim is to provide open and flexible tools for the mode recognition and tonic identification tasks, which can be applied to different music cultures while allowing the users to optimize the parameters easily according to the characteristics of the studied music.

This toolbox also includes the implementations of pitch and pitch-class distributions (Section 5.5)[33] and also the stable pitch and pitch-class extraction step (Section 5.6)[34] and hence provide the essential tools for several relevant tasks such as tuning and intonation analysis (Section 5.9 and 6.11). Moreover, `MORTY` has been recently used as a benchmark against novel methodologies proposed for mode recognition in Hindustaion and Carnatic music (Gulati, Serrà, Ishwar, et al., 2016; Gulati, Serrà, Ganguli, et al., 2016) (Appendix B.7).

In addition to the toolbox and experimentation code, the dataset (Section 5.7.4), the experiments (Section 5.7.3) and the results (Sec-

[30] Excluding the commercial audio recordings, which cannot be generally made public due to copyright laws.

[31] `https://github.com/altugkarakurt/morty`

[32] `https://www.gnu.org/licenses/agpl-3.0.en.html`

[33] `https://github.com/altugkarakurt/morty/blob/v1.2.1/morty/pitchdistribution.py`

[34] `https://github.com/altugkarakurt/morty/blob/v1.2.1/morty/pitchdistribution.py#L231-L265`

tion 5.7.5) are in public domain.[35]

Experimental Setup

In the experiments we use stratified 10-fold cross validation. Table 5.2 gives a combination of the parameters used in the experimental setup. We use grid search, to find the optimal parameters for OTMM. We use `ATL-MEL`$_f$ for predominant melody extraction (Atlı et al., 2014). The parameter combinations where the bin size $b\left(\hat{\boldsymbol{H}}\right)$ is greater than or equal to 3 times the kernel width $\sigma\left(\hat{\boldsymbol{H}}\right)$ are omitted. We also conduct experiments using the raw distributions, without smoothing. When the training model consists of a "single" distribution per mode, the number of neighbors, is always taken as 1 as each label is represented by a single data point. The minimum peak ratio, $\delta(\boldsymbol{H})$, is only used in tonic identification and joint estimation tasks. The optimal value of the minimum peak ratio is found separately (Section 5.7.5).

For mode recognition, we mark the classification as $True$, if the estimated mode $\mu^{(a)}$ and the annotated mode $\mathfrak{m}^{(a)}$ for a recording (a) are the same. The tonic octave in the orchestral performances of OTMM is ambiguous as each instrument plays the melody in their own register. Therefore, we aim to evaluate the tonic pitch class and calculate the shortest octave-wrapped cent distance $\blacktriangle(\kappa^{(a)}, \mathfrak{t}^{(a)})$ between the estimated $\kappa^{(a)}$ and the annotated tonic $\mathfrak{t}^{(a)}$ (Equation 5.3). If the distance is less than 25 cents, we consider the tonic as correctly identified. In the case of joint estimation, we require both the tonic and makam estimates to be correct.

For each fold, we compute the accuracy, which is the number of correct estimations divided by the total number of testing data. In Section 5.7.5, we report the average accuracies of the folds for each parameter combination. We also compare the tonic identification results obtained in the tonic identification and joint estimation tasks with the results obtained from the the last note detection method (Atlı et al., 2015) (Section 5.7.2). For all results below, the term "significant" refers to statistical significance at the $p = 0.01$ level as determined by a multiple comparison test using the Tukey-Kramer statistic.

[35] `http://compmusic.upf.edu/node/319`

Table 5.2: The summary of the tasks, features, training models and parameters used in the experiments.

Symbol	Name	Values / Methods	Comment
	task	mode, tonic, joint	
ϱ	predominant melody	$\texttt{ATL-MEL}_f$	extraction method specialized for OTMM
$\hat{\boldsymbol{H}}$	distribution	PD, PCD	
$\boldsymbol{\mathcal{T}}$	type of the training model	single, multi	number of distributions per mode used in (Gedik & Bozkurt, 2010; Chordia & Şentürk, 2013)
$b\left(\hat{\boldsymbol{H}}\right)$	bin size	$7.5, 15, 25, 50, 100$ cents	
$\sigma\left(\hat{\boldsymbol{H}}\right)$	kernel width	"no smoothing" & $7.5, 15, 25, 50, 100$ cents	Combinations with $b\left(\hat{\boldsymbol{H}}\right) \geq 3\sigma\left(\hat{\boldsymbol{H}}\right)$ are omitted.
$\diamondsuit$	distance or dissimilarity	L_1, L_2, L_3, Bhattacharyya, $1-$intersection, $1-$cross_correlation	$1-$intersection and $1-$cross_correlation are dissimilarities computed from the namesake similarity measures
k	number of nearest neighbors	$\{1, 3, 5, 10, 15\}$	for the "single" distribution per mode training model, the value is fixed to 1
$\delta(\boldsymbol{H})$	minimum peak ratio	$[0, 1]$	optimal found by a separate grid-search with a step of 0.05, is not used in makam recognition

To find an optimal for the minimum peak ratio, $\delta(\boldsymbol{H})$, we compute all distributions of each recording in the dataset using all the combinations of the bin sizes and the kernel widths given in Table 5.2. Then, we detect the peaks in each pitch distribution using a minimum peak ratio, from 0 (no threshold) to 1 (only keeping the highest peak). For each value of the minimum peak ratio, we note the number of distributions which has the annotated tonic among the peaks ("tonic hits") and the total number of peaks obtained from each distribution. All of the scripts, computed features, experiments and results are shared online for reproducibility purposes.[36]

[36] https://github.com/sertansenturk/makam_recognition_experiments/tree/dlfm2016

5.7.4 Dataset

In (Gedik & Bozkurt, 2010), the makam recognition method was evaluated on 172 solo audio recordings in 9 makams. To the best of our knowledge, this dataset represents the biggest number of recordings that has been used to evaluate makam recognition task, so far. As explained by the authors, these recordings were selected from the performances of "indisputable masters," and therefore they are musically representative of the covered makams. Nevertheless, the results are not reproducible as the dataset is not public.

The tonic identification method proposed in (Bozkurt, 2008) was evaluated using 150 synthesized MIDI files plus 118 solo recordings. Similar to (Gedik & Bozkurt, 2010) the data is not publicly available. To the best of our knowledge, the only open tonic identification datasets have been compiled under the CompMusic project (Section 3.2.4). The first one is provided in (Şentürk et al., 2013) (which will be explained in detail in Section 6.4) and it consists of 257 audio recordings. The second and the bigger test dataset is provided in (Atlı et al., 2015), consisting of 1093 recordings.[37] The recordings in both of the datasets are identified using MBIDs. Nevertheless, the features extracted from the audio recordings are not provided in either dataset. Therefore, the results are not straighforward to reproduce.

Considering the lack of open test datasets for makam recognition and the drawbacks of the available tonic identification datasets, we have gathered a test dataset of audio recordings with annotated makam and tonic, called the *Ottoman-Turkish makam recognition dataset*.[38] The dataset covers 20 commonly performed makams[39] and it is composed of 1000 audio recordings. Following our constraint in the problem definition (Section 5.7.1), a single makam is performed in each recording (i.e. there are 50 recordings per makam). This dataset is currently the largest and the most compre-

[37] The datasets are hosted in `https://github.com/MTG/turkish_makam_tonic_dataset/releases/`

[38] `https://github.com/MTG/otmm_makam_recognition_dataset/tag/v1.0.0`

[39] i.e. Acemaşiran, Acemkürdi, Bestenigar, Beyati, Hicaz, Hicazkar, Hüseyni, Hüzzam, Karcığar, Kürdilihicazkar, Mahur, Muhayyer, Neva, Nihavent, Rast, Saba, Segah, Sultanıyegah, Suzinak and Uşşak

hensive dataset for the evaluation of automatic makam recognition. Moreover, it is comparable to the aforementioned dataset provided in (Atlı et al., 2014) for the evaluation of tonic identification methodologies.

Similar to (Atlı et al., 2015) and (Şentürk et al., 2013), the recordings in the dataset are labeled with MBIDs. The tonic frequency of each recording is annotated manually by marking the tonic frequency using the visual interface of Makam Toolbox. The tonic frequency is adjusted by synthesizing and listening the marked frequency synchronous to the audio playback using the same toolbox. The annotations are cross checked by at least two annotators. We also provide the predominant melodies extracted from the audio recordings using $\texttt{ATL-MEL}_f$ for reproducibility purposes.

Similar to (Gedik & Bozkurt, 2010), the dataset is intended to be musically representative of OTMM. To achieve this, we selected the recordings of acknowledged musicians from the CompMusic makam corpus (Uyar et al., 2014), which is currently the most representative music corpus of OTMM aimed at computational research. The dataset covers contemporary and historical, monophonic and heterophonic recordings, as well as live and studio recordings. Some of the recordings have non-musical sections, such as clapping at the end of live recordings, announcements or scratch and hissing sounds (e.g. in historical recordings). This diversity gives us the opportunity to test the methods in a much more challenging environment, which has not been completely addressed in previous research (Gedik & Bozkurt, 2010).

5.7.5 Results

To find an optimal for the minimum peak ratio, $\delta(\boldsymbol{H})$, we compute the PD and PCD for all recordings in the test dataset using 7.5 and 15 cent bin size and all kernel widths given in Table 5.2, resulting in 24 distributions $\times 1000$ recordings $= 24000$ distributions. We detect the peaks in the computed distributions by changing the minimum peak ratio from 0 (select all peaks) to 1 (select the highest peak), and note whether the annotated tonic is among the detected peaks. For each minimum peak ratio, we divide the number of distributions with the tonic annotation detected as a peak and the total number of computed distributions (24000). Note that this

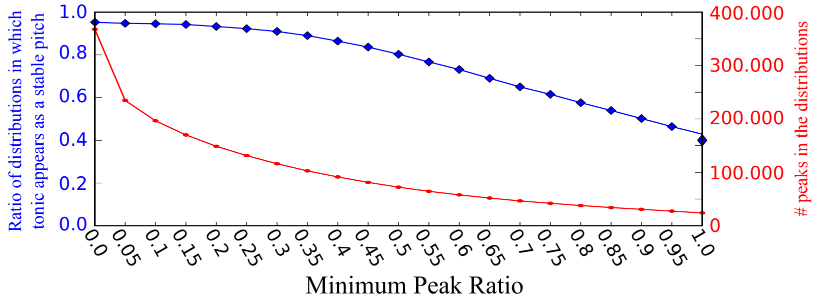


Figure 5.10: Total number of peaks and the ratio between the number of tonic hits and number of all distributions.

ratio is not 1 even when $\delta(\boldsymbol{H})$ is selected as 0. This is because of the possibility that the tonic masked by the neighboring peaks when the selected kernel is too wide. We also count the total number of peaks detected from all distributions for each minimum peak ration. By inspecting the Figure 5.10, we observe that the probability of finding the tonic among the peaks is very high for minimum peak ratios less than 0.4 in the expense of an exponential increase in the tonic candidates (peaks) and hence in the processing time. Since our scenario can tolerate a moderate increase in processing time, we select the minimum peak ratio, $\delta(\boldsymbol{H})$, as 0.15.

Table 5.3 shows the best results obtained after grid search. For mode recognition, multi-distribution per mode model yields an accuracy of 71.8% with the best parameter set while highest accuracy using single distribution per mode is 38.7%. For tonic identification multi-distribution per mode performs with accuracy above 95% in 20 parameter sets and above 90% accuracy in 299 parameter sets out of 1440 experiments, where the highest accuracy obtained is 95.8%. Hence, the method is robust to a variety of parameter selections for tonic identification. On the other hand, single distribution per mode model yields 89.8% accuracy with the best parameter set. For joint estimation the multi-distribution per mode model performs with 63.6% accuracy (86.1% tonic identification and 65.2% makam recognition accuracy) in the best configuration, while single distribution yields 27.6% (71.0% tonic identification and 28.6% makam recognition accuracy). For all three considered tasks, the optimal choices for distribution, distance/dissimilarity and training model are PCD, Bhattacharyya distance and multi-

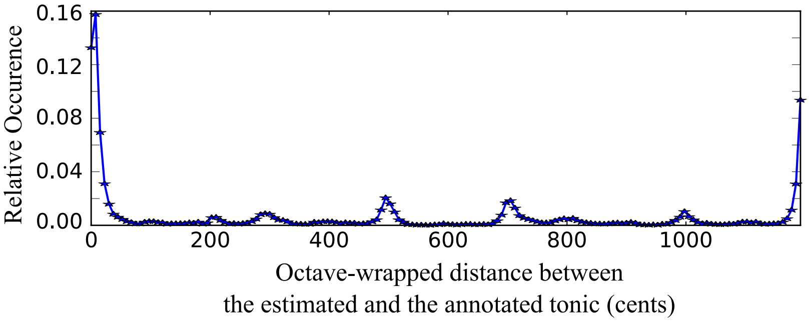


Figure 5.11: The distribution of octave-wrapped distances between the estimated and annotated tonic for all parameter sets with 7.5 cent bin size.

distribution per mode.

Table 5.3: The best parameter sets for each task. For all tasks PCDs using Bhattacharyya distance and traning multiple distributions per mode gives the best results.

Task	$\sigma\left(\hat{\boldsymbol{H}}\right)$	$b\left(\hat{\boldsymbol{H}}\right)$	**k**	**Accuracy**
Tonic	7.5	15	3	**95.8%**
Makam	25	25	10, 15	**71.8%**
Joint	15	7.5	15	**63.6%**

The method proposed in (Atlı et al., 2015) obtained 79.9% tonic identification accuracy on our dataset. The best tonic identification accuracy using PDs and single-distribution per mode as proposed in (Bozkurt, 2008) is 49.8%. Multi-distribution per mode method using PCDs outperforms both methods whether the makam is known (95.8% accuracy with the best configuration) or not (91.5% tonic accuracy in joint estimation with the best configuration) even with the majority of sub-optimal parameter sets. Figure 5.11 shows the distribution of the octave-wrapped cent distance (Equation 5.2) between the estimated and the annotated tonic for each test and for all the parameter sets with 7.5 cent bin size.

These experiments revealed that certain parameter selections significantly improve or diminish the methods' performances:

- $\boldsymbol{\mathcal{T}}$**:** Multi-distribution training model (Chordia & Şentürk, 2013) performs significantly better than single-distribution training model (Gedik & Bozkurt, 2010).
- $\hat{\boldsymbol{H}}$**:** PCD significantly outperforms PD.
- $b\left(\hat{\boldsymbol{H}}\right)$**:** Smaller bin size yields better results, however there is no significant distinction between 7.5, 15 and 25 cent bin sizes. Note that these bin sizes significantly outperform 50 and 100 cent bin sizes.
- $\sigma\left(\hat{\boldsymbol{H}}\right)$**:** The 7.5, 15 and 25 cent kernel widths significantly improves the accuracy of the models compared to 50 and 100 cent kernel widths. No smoothing performs slightly (insignificantly) worse than 7.5, 15 and 25 cent kernel widths. However, processing the distribution without smoothing is substantially slower due to the peak detection step.
- $\diamondsuit$**:** Using multi-distribution training model and PCDs, Bhattacharyya distance always yields the highest accuracy. It is significant for all cases except using either $1-$intersection or $L1$ in tonic identification.
- $\boldsymbol{k}$**:** Increasing the number of nearest neighbors increases the accuracy. Nevertheless, the increase does not make a significant impact except $k = 1$, which performs significantly worse than higher k values.
- $\delta(\boldsymbol{H})$**:** In the tonic identification task, the true tonic is typically among the detected peaks for minimum peak ratios below 0.4. Values smaller than 0.1 increases the processing time without any meaningful improvement in tonic identification accuracy.

5.7.6 Discussion

The drawback of the pitch distribution based methods is that they do not consider the temporal characteristics. When we inspect the results obtained from the experiments in 5.7.3, it is observed that the confusions (Figure 5.12) are mainly between makams, which either have very similar intervals in their scale or contain similar

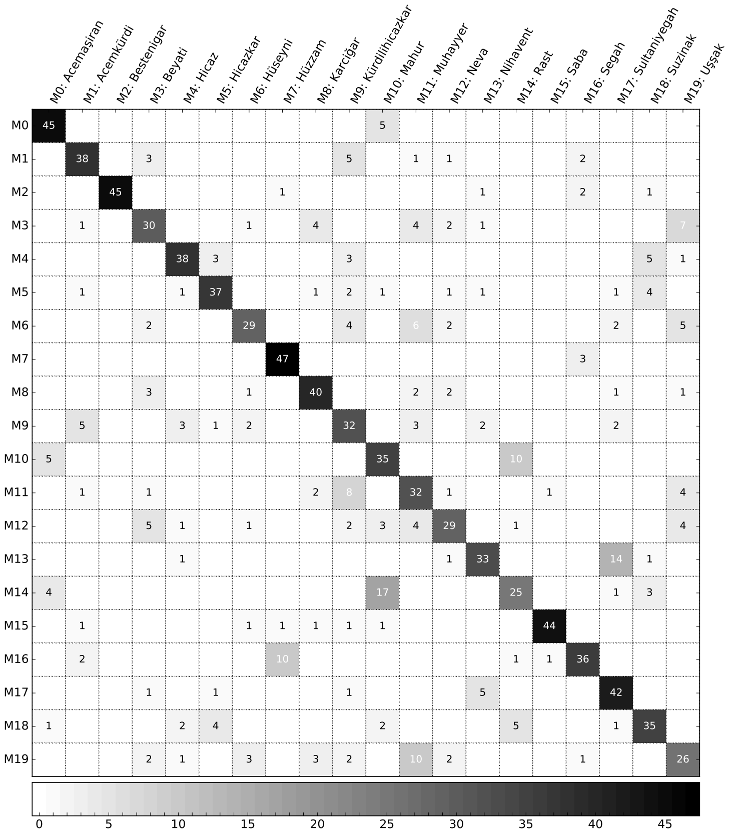


Figure 5.12: Confusion matrix of the best performing makam recognition experiment (Table 5.3).

sets of pitches. Similarly in (Gulati, Serrà, Ishwar, et al., 2016), the proposed method was better in classifying phrase-based ragas, while our method was better at classifying scale based ones (Appendix B.7).

In (Atlı et al., 2015), we showed that the last note detection method outperforms the tonic identification method in (Bozkurt, 2008) (i.e. using PDs with single-model per mode) for OTMM. Our results validate these findings (the best is accuracy is 49.8 as stated in Section 5.7.5). Nevertheless, we show that using PCDs with multi-model per mode is superior to both methods even when the makam of the recording is not known and even if the makam is found erroneously in the joint estimation process.

While the estimated tonic is typically around the annotation

(Figure 5.11), the main confusion occurs around the fourth, fifth and seventh of the tonic, which typically act as the melodic centers and/or anchor points in the melodic progression (Özkan, 2006).

For all the tasks defined in Section 5.7.1, we suggest using multi-distribution models approach with PCD and Bhattacharyya distance. If the estimation accuracy is a top priority, we suggest choosing a small $b\left(\hat{\boldsymbol{H}}\right)$, $\sigma\left(\hat{\boldsymbol{H}}\right)$ (7.5 or 15 cents) and $\delta(\boldsymbol{H})$ (0.15) as these parameters yield high accuracies. For applications requiring computational efficiency (e.g. mobile applications) or fast operation (e.g. real-time estimation), $b\left(\hat{\boldsymbol{H}}\right)$, $\sigma\left(\hat{\boldsymbol{H}}\right)$ (25 cents) and $\delta(\boldsymbol{H})$ (0.4) can be bigger, since reduced feature dimensions would substantially decrease the computational complexity. The number of neighbors may be chosen as any value higher than 1.

5.7.7 Summary

In this section, a generalized methodology based on (Bozkurt, 2008; Gedik & Bozkurt, 2010; Chordia & Şentürk, 2013) is presented. The methodology is implemented as an open toolbox for mode recognition and tonic identification called `MORTY`. It is designed with flexibility in mind such that it can be easily modified and optimized to analyse large audio corpora. The implementation is evaluated on the largest makam recognition dataset of OTMM. The generalized method outperformed the state-of-the-art methodologies proposed for makam recognition (Gedik & Bozkurt, 2010) and tonic identification (Bozkurt, 2008; Atlı et al., 2014). The toolbox has also been used to benchmark two novel mode recognition methodologies proposed for Indian art musics (Appendix B.7).

`MORTY` is also used as a part of our makam music analysis toolbox[40] in several tasks such as pitch and pitch-class distribution computation (Section 5.5), tuning analysis (Section 5.9) and melodic progression analysis (Section 5.10). The usage and implementation will be explained more in Section 5.11) and Appendix C, respectively. In the future, we plan to apply dimension reduction and hashing techniques to summarize the features and speed up the classification for real-time mode and tonic estimation on short audio

[40] https://github.com/sertansenturk/tomato

fragments. We would also like to incorporate the (Gulati, Serrà, Ganguli, et al., 2016), which has outperformed our methodology in mode recognition on Carnatic and Hindustani musics. We also hope that MORTY may be useful as a general tool for tonic identification, mode recognition and tuning analysis applied on different modal music traditions.

5.8 Transposition

To obtain the transposition (ahenk) of an audio performance, the tonic symbol of the makam of the recording is read by referring to a dictionary.[41] The theoretical frequency (according to the intervals defined by AEU theory) $₺_{bolahenk}$ of the tonic in Bolahenk (the default transposition of OTMM performances) is also noted.[42] Then, the octave-wrapped cent distance $\triangle(\kappa, ₺_{bolahenk})$ from κ to $₺_{bolahenk}$ is computed (Equation 5.2). Finally, the transposition is matched by referring to the interval in Table 5.4, which contains the computed distance.[43] Note that the ”Bolahenk Nısfiye” ahenk, which is an octave higher than Bolahenk, is omitted due to the ambiguity of tonic octave in heterophonic recordings (Section 2.1).

5.9 Tuning

To analyze the tuning of each note performed in an audio fragment, we implemented the methodology explained in (Bozkurt et al., 2009).[44] The method first obtains the set of stable pitches $\mathbf{\Phi}_P$ performed in an audio fragment (a) by applying peak detection

[41] `https://github.com/sertansenturk/ahenkidentifier/blob/v1.5.0/ahenkidentifier/data/tonic.json`

[42] e.g. A4 $\approx$ 329.6 Hz, if the tonic symbol of the makam is A4. Readers are reminded that the “typical” performance tuning of Western classical music (A4 = 440 Hz) is a fourth higher than Bolahenk (Section 2.1).

[43] The implementation is available at `https://github.com/sertansenturk/ahenkidentifier`.

[44] This work was done by Hasan Sercan Atlı and me between January and March 2016. We have equal contribution on implementing the methodology. I have also contributed to the code with bug fixing, refactoring, documentation and deployment. Bilge Miraç Atıcı has also contributed to the initial stages of the development and also with bug fixing.

Table 5.4: The transpositions, the corresponding octave-wrapped cent distance intervals and the theoretical center of the pitch-class if the tonic symbol is G4.

Ahenk	$\triangle(\kappa, \mathfrak{k}_{bolahenk})$	**Center when the tonic is G4**	
Bolahenk	$[-50, 50)$ cents	293.67 Hz	
Davut-Bolahenk Mabeyni	$[50, 150)$ cents	311.13 Hz	
Davut	$[150, 250)$ cents	329.63 Hz	
Şah	$[250, 350)$ cents	349.23 Hz	
Mansur-Şah Mabeyni	$[350, 450)$ cents	370.00 Hz	
Mansur	$[450, 550)$ cents	392.00 Hz	
Kız-Mansur Mabeyni	$[550, 650)$ cents	415.31 Hz	$\times 2^n, \forall n \in \mathbb{Z}$
Kız	$[650, 750)$ cents	440.01 Hz	
Yıldız	$[750, 850)$ cents	466.17 Hz	
Müstahsen	$[850, 950)$ cents	493.89 Hz	
Sipürde	$[950, 1050)$ cents	523.26 Hz	
Bolahenk-Sipürde Mabeyni	$[1050, 1150)$ cents	554.38 Hz	

on the pitch distribution as explained in (Section 5.6). The stable pitches are then normalized (Equation 5.1) with respect to the tonic frequency κ (Section 5.7) and the performed scale degrees $\hat{\phi}^{\kappa}_{P,i} \in \hat{\mathbf{\Phi}}^{\kappa}_{P}$ (in cents) are obtained.

In parallel, the note symbols in the scale of the performed makam is inferred from the key signature of the makam and extended to ± two octaves. The note symbols are mapped to the theoretical scale degrees according to the AEU theory (Özkan, 2006) (e.g. if the tonic symbol is G4, the scale degree of A4 is 9 Hc $\approx$ 203.8 cents).

Next, the performed scale degrees are matched with the theoretical scale degrees using a threshold of 50 cents (close to 2.5 Hc, which is reported as the optimal by Bozkurt et al. (2009)). If a performed scale degree is close to more than one theoretical scale degree (or vice versa), we only match the closest pair. As a trivial addition to (Bozkurt et al., 2009), we re-map the theoretical scale degrees to the note symbols and obtain the *stable pitch - note symbol* pairs, i.e. $\langle \phi_i, n_i \rangle$s. We also store the theoretical scale degree and the performed scale degree of each match.[45]

Figure 5.13 shows the extracted tuning over the pitch distribution of an audio recording in Hüseyni makam.[46] The frequency of

[45] Our implementation is available at https://github.com/miracatici/notemodel.

[46] http://musicbrainz.org/recording/8b8d697b-cad9-446e-ad19

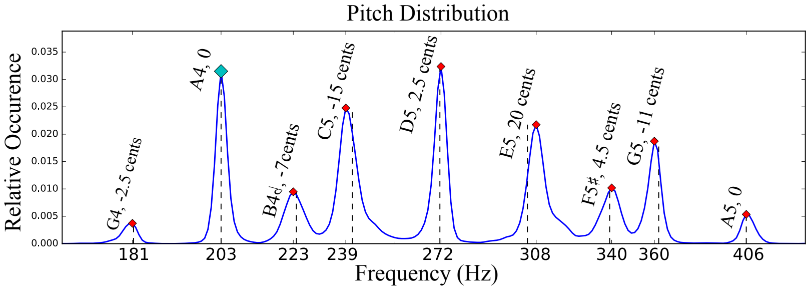


Figure 5.13: The tuning extracted from an audio recording in Hüseyni makam.

each stable note is shown on the x-axis. The vertical dashed lines indicate the frequencies of the notes according to the theoretical intervals. The matched note symbol and the deviation from the theoretical scale degree of each stable pitch is displayed right next to the corresponding peak on the PD. It can be observed that the some of the notes (esp. çargah and hüseyni notes) substantially deviate from the AEU theory.

Note that the method explained in (Bozkurt et al., 2009) is limited to obtaining the tuning of the notes in the extended scale. Moreover, there is limited information about the intonation of the performed notes that can be retrieved from analysing the pitch distributions alone.[47] In Section 6.11, a score-informed method is proposed, which is able to capture the tuning and intonation of the majority of the performed notes.

5.10 Melodic Progression

Bozkurt (2015) has proposed two similar models for analyzing the melodic progression (seyir) of music scores and audio recordings, respectively. For symbolic analysis, the note sequence in the score is divided into small chunks. The note sequence in each chunk is synthesized (Section 4.2.2) by converting the note symbols to scale degrees according to the intervals defined by the AEU theory. Then the relative occurrence of the scale degrees and the mean of the scale degrees is computed for each chunk. For audio anal-

-5e85a36aa253

[47] e.g. the shape and spread of the peaks in the distributions

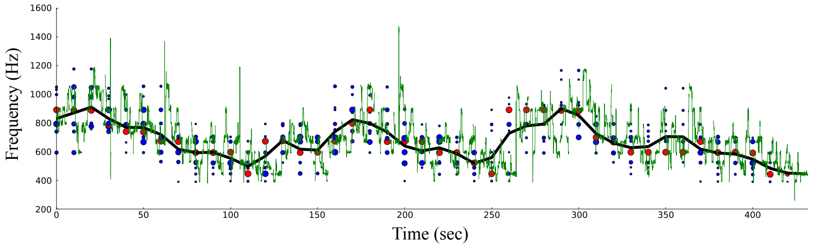


Figure 5.14: The predominant melody and melodic progression feature of an audio recording.

ysis, the predominant melody is divided into chunks and a pitch distribution is computed for each chunk. Finally the mean of the predominant melody of each chunk is computed. Bozkurt (2015) computes these representations from several sets of music scores or audio recordings grouped with respect to their makam to observe a "generalizable" melodic progression for each of the studied makams.

To analyse the melodic progressions in an audio fragment, these two approaches are combined by first extracting a predominant melody $\boldsymbol{\varrho}$ and dividing it to chunks with a certain frame size and overlap ratio. Then a PD $\boldsymbol{H}_P$ (Section 5.5) is computed for each chunk and the set of stable pitches $\boldsymbol{\Phi}_P$ are extracted from the PD of each chunk (Section 5.6).[48]

So far the melodic progression is used for visualization in the recording pages of Dunya-makam (Section 7.1.2). For consistency in visualization, we divide an input audio recording into 40 chunks with a 50% overlap. Figure 5.14 shows the predominant melody (in green) extracted from an audio recording and along with the melodic progression. The black line indicates the mean of the predominant melody in each chunk. The dots show the location of the stable pitches. The radius of the dots are proportional to the height of the corresponding peak in the PD computed for the chunk and the red dot corresponds to the stable pitch extracted from the highest peak of the PD in its chunk. Note that the computed PDs are not displayed to avoid cluttering the Figure.

[48] The implementation is available at `https://github.com/sertansenturk/seyiranalyzer`.

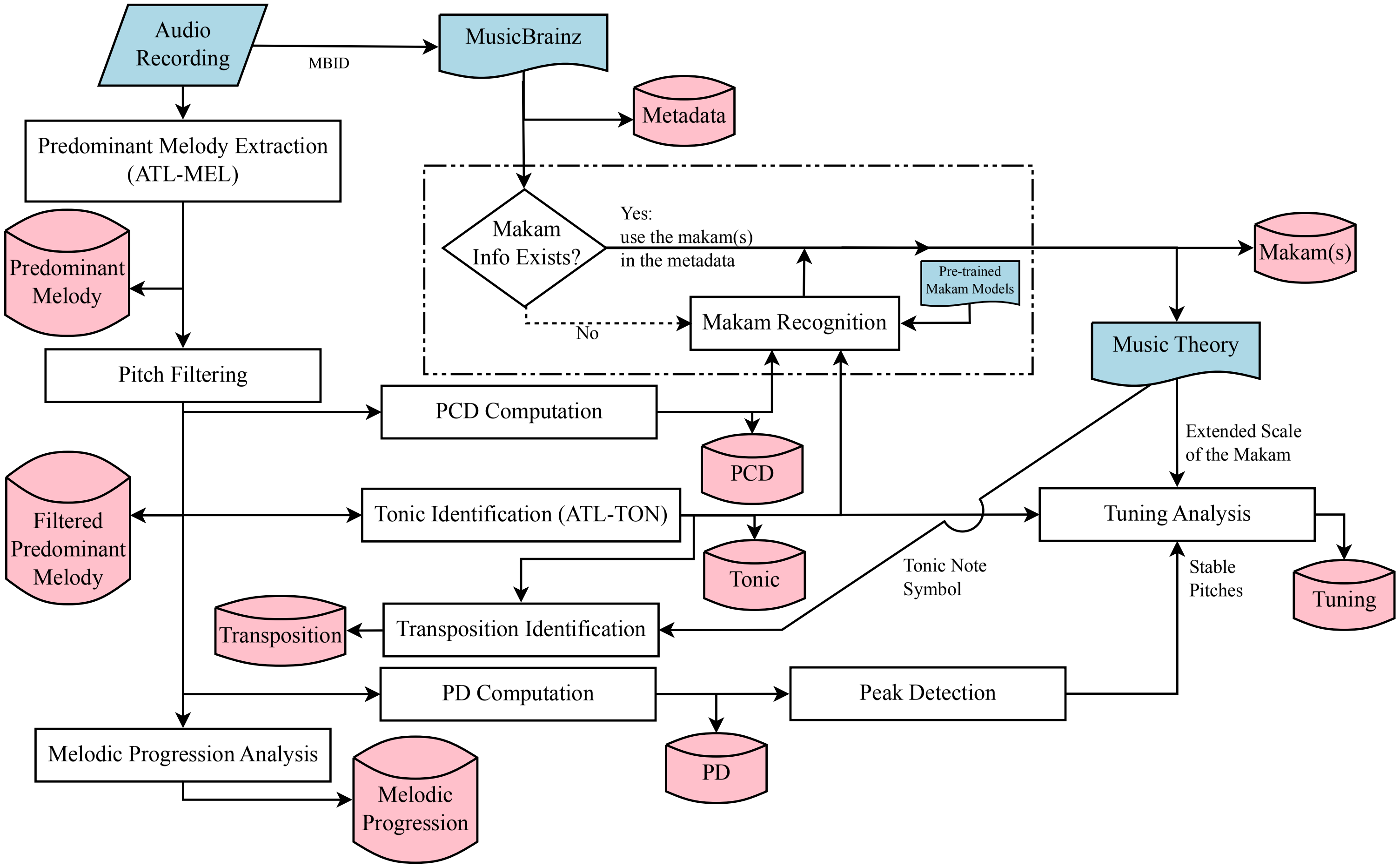


Figure 5.15: The audio analysis process.

5.11 Combining Audio Analysis Methodologies

The methodologies described throughout the Chapter are implemented in Python and integrated into an audio analysis sub-package in **T**urkish-**O**ttoman **M**akam (M)usic **A**nalysis **TO**olbox (`tomato`).[49] If a type of information is alreadyFor example, if the key signature of the makam of a recording is not known (Section 3.1.3), the stable pitches will not be matched with the note symbols during tuning analysis. The algorithms may also be executed together in a workflow as shown in Figure 5.15. In the combined analysis workflow, the makam recognition step is skipped, if the makam information already exists in the metadata.

For a more detailed description of the implementations, please refer to Appendix C.

5.12 Automatic Description of the CompMusic-Makam Audio Collection

Using the combined audio analysis methodologies (Section 5.11), an automatic description of the CompMusic OTMM audio collection is obtained. Figure 5.16 shows the overview of the description.

The automatic description is used to facilitate the navigation and discovery on the audio collection itself (Section 7.1.2). In addition, the description could allow to study the performance characteristics on a sizable amount of data. As an example, Figure 5.17 and Figure 5.18 show the distribution of automatically identified tonic pitch-classes and transpositions from the CompMusic OTMM audio collection using `ATL-TON`. It can be observed that the most popular ahenks are bolahenk, mansur, kız and sipürde. In addition, the spread shows that the tonic is not necessarily tuned with respect to a standard reference frequency such as 440 Hz.

[49] `https://github.com/sertansenturk/tomato/blob/v0.9.1/tomato/audio/audioanalyzer.py`

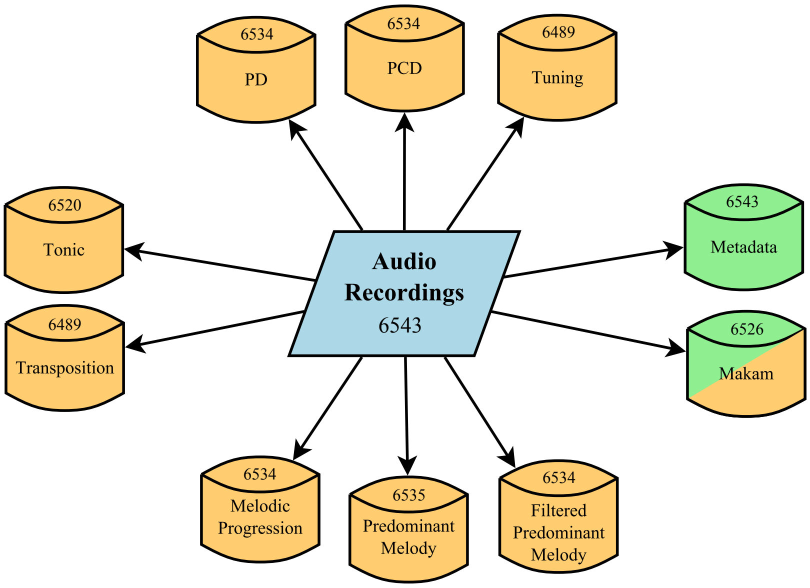


Figure 5.16: An overview of the description of the CompMusic OTMM audio collection. The numbers in the boxes indicate the number audio recordings for which the relevant entity is extracted. The metadata fetched from MusicBrainz is shown in green and the features obtained by automatic analysis are shown in orange. The makam information is shown in both green and orange because it obtained either from the metadata or automatically (Section 5.11).

Note that `ATL-TON` achieved around 80% accuracy on the Makam Recognition Dataset (Section 5.7.5, hence these distributions consist of a significant amount of erroneous information. This argument can be extended to other audio features, which are explained throughout this Chapter. In the next Chapter, improvements over the automatic description will be discussed using methods based on joint audio and score analysis.

5.13 Conclusion

In this Chapter, an overview of the melodic analysis tasks applied to audio recordings of OTMM is presented. The Chapter covers most of the automatic analysis tasks discussed in (Bozkurt, Ayangil, & Holzapfel, 2014), namely predominant melody extraction, pitch

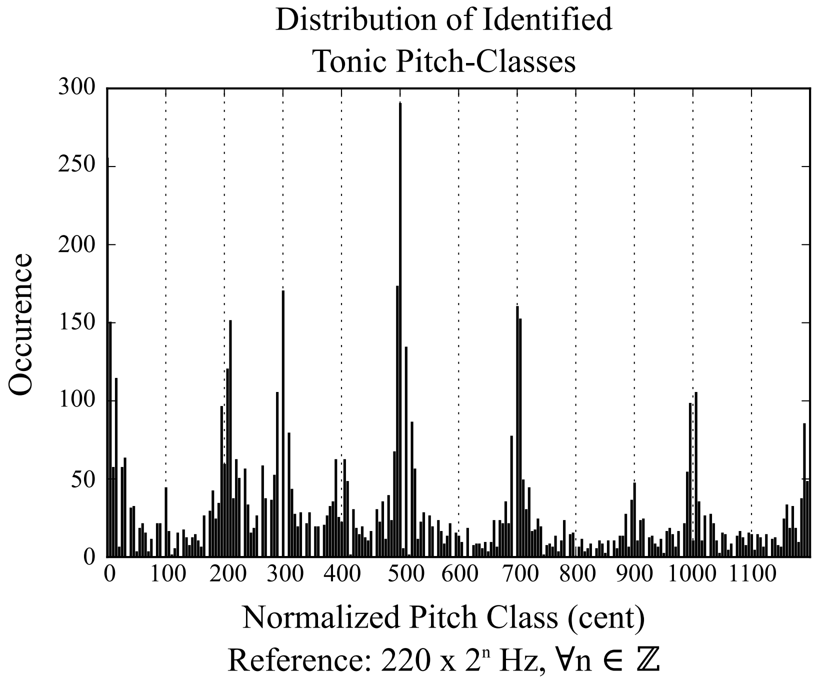


Figure 5.17: The distribution of tonic pitch classes in the CompMusic OTMM audio collection using ATL-TON.

and pitch-class distributions computation, stable pitch and pitch-class computation, makam recognition, tonic and transposition identification, tuning analysis and melodic progression computation. One notable example, which is not left out of this Chapter is automatic transcription. The readers are referred to (Benetos & Holzapfel, 2015) for a recent study on this problem.

Some of the existing methodologies presented in this Chapter have been improved for better performance on audio recordings of OTMM. Among these methods, ATL-MEL$_f$ is currently the state-of-the-art in predominant melody extraction. In addition, we have generalized the distribution-based tonic and makam estimation methodologies previously proposed for OTMM (Gedik & Bozkurt, 2010) and Hindustāni music (Chordia & Şentürk, 2013). The generalized method outperforms previously proposed methodologies in tonic identification (Atlı et al., 2015) and makam recognition (Gedik & Bozkurt, 2010).

To facilitate future development, the implementations of the

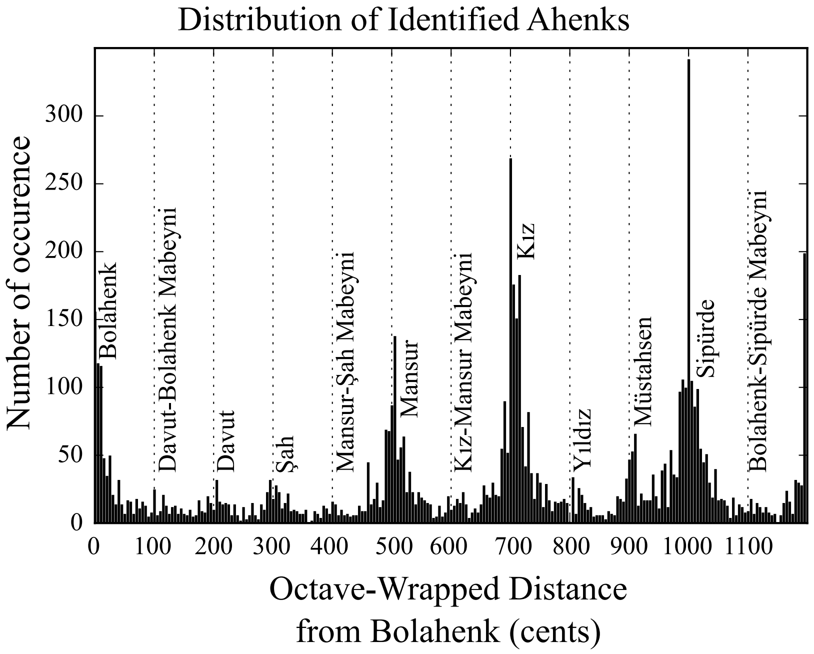


Figure 5.18: The distribution of transpositions in the CompMusic OTMM audio collection using `ATL-TON`.

methodologies described in this Chapter are distributed openly in `tomato`. In addition, these algorithms are used to analyse the CompMusic OTMM audio collection and an automatic description of the collection is obtained as a result. Many features constituting the description will be improved later as a result of the joint audio and score analysis (Chapter 6).

Several of the methodologies described in this Chapter (such as `ATL-MEL`$_f$ and `ATL-TON`) rely on rule-based schemes. These approaches were taken due to the lack of data in the initial stages of the CompMusic project. The resulting description obtained from the automatic analysis could now be used to develop more sophisticated and robust approaches based on machine-learning. The description of tonal space may also be improved by replacing the PCDs with *time-delayed melody surface* (Gulati, Serrà, Ganguli, et al., 2016), which is already shown to outperform PCD on the rāga/rāg recognition task in IAM (Section B.7).

Chapter 6

Joint Audio-Score Analysis

The most relevant representations of music are notations and audio recordings, each of which emphasizes a particular perspective and promotes different approximations in the analysis and understanding of music. *Linking* these two representations and analyzing them jointly should help to better study many musical facets by being able to combine complementary analysis methodologies. Parallel information extracted from score and audio recordings may facilitate many computational tasks such as version detection (Arzt et al., 2012), source separation (Ewert & Müller, 2012), intonation analysis (Devaney et al., 2012; Abesser et al., 2016) and automatic accompaniment (Cont, 2010).

In order to develop accurate alignment methods, we have to take into account the specificities of a given type of music. In this Chapter presents an audio-score alignment methodology, which is designed to address several challenges brought by the musical characteristics of OTMM performances such as transpositions of tonic, tuning and intonation deviations, and heterophony. In addition the methodology is robust to structural differences, and melodic additions, insertions and omissions between the music scores and audio recordings.

The alignment procedure is based on linking a fragment selected from the music score with the audio recording (Section 6.3).

This step is designed to find inexact matches due to the characteristic differences between the melodic representations of these information sources. Fragment linking is used to jointly identify the tonic frequency (Section 6.4) and estimate the tempo (Section 6.5). It is also used to automatically identify the performed music compositions in an audio collection, and vice versa (Section 6.6). As a result of composition identification, the performances and compositions in a music corpus could be linked with each other.

To overcome the structural differences (e.g. section repetitions, improvisations) between the audio recordings and music scores, a bottom-up approach is used based on fragment linking (Section 6.7). Instead of attempting to align the complete music score with the audio recording, the methodology estimates candidate locations in the audio performance for each the musical section in the score. Then, the best section-level alignment is inferred from the candidates. In parallel, a finer alignment is applied between the linked sections in the audio recording and music score to obtain the note-level alignment (Section 6.8).

As an output of the alignment process, the performed notes and sections in the audio recordings are obtained. In addition, these events are linked to the relevant notes and sections in the music score. The linked score information could be further exploited to infer additional (time-aligned) features such as lyrics and rhythmic elements (measure, usul etc.) (Section 6.9). The aligned notes are used to refine several audio features such audio predominant melody, pitch distribution, pitch-class distribution and melodic progression (Section 6.10), and to construct more-informed tuning and intonation models (Section 6.11).

The aforementioned steps are incorporated into a single workflow (Section 6.12), which is used to extend and complement the automatic description of the CompMusic OTMM corpus. The results show that the joint audio-score analysis does not only improve the automatic description compared to the state-of-the-art audio analysis scheme (Chapter 5), but it also brings a simpler solution for many computational tasks, which would require sophisticated computational methodologies otherwise.

The contributions presented in this Chapter may be summarized as:

- A novel audio-score alignment approach for OTMM, which is designed to handle culture-specific challenges brought by the musical characteristics of OTMM
- Robust score-informed audio analysis methodologies for numerous computational tasks such as tonic identification, tempo estimation, composition identification, predominant melody filtering, and tuning and intonation analysis.
- Open and easy-to-use implementations of the joint audio-score analysis methodologies.
- Open datasets to evaluate these methodologies.
- Automatic description of the CompMusic OTMM corpus obtained from joint audio-score analysis. The description encompasses approximately $18,000$ linked sections and $750,000$ notes, which correspond to more than 85 hours of time-aligned audio data.

Now, I proceed to introduce the basic terminology used thoughout the Chapter.

6.1 Nomenclature

We define *audio-score alignment* as *synchronisation of the musical events in the score of a composition with the corresponding events in the audio recording of the same composition*. In this Chapter, two levels of granularity are considered in the alignment: **1)** Structure (section) level, **2)** Note level. Our method addresses the some of main challenges of computational analysis of OTMM such as transpositions, structural differences and tuning deviations.

1. Let $\bar{\mathbf{N}}^{(b)}$ the note sequence in the music score (b). Each $\bar{n}_j^{(b)} \in \bar{\mathbf{N}}^{(b)}$ (where $j \in \left[1 : |\bar{\mathbf{N}}^{(b)}|\right]$) consists of a $\left\langle n_j^{(b)}, d\left(n_j^{(b)}\right)\right\rangle$ tuple, the elements of which represent the note symbol, note duration associated with the note, respectively.

   Throughout this Chapter, the *time* and *duration* of a "score fragment" (defined in the next point) does not refer to symbolic time and duration, respectively. Instead, these symbolic values are converted to seconds by referring to a certain tempo in bpm (e.g. the duration of a eighth note ♪ according to the tempo ♩ $=$ 120 bpm is 0.25 seconds). Unless stated explicitly, the nominal tempo indicated in the music score is used for conversion. The nominal tempo of a music score (b) is denoted as $\tau^{(b)}$.

2. Let $\bar{f}^{(x)}$ be an arbitrary fragment selected from a music score or an audio recording (x). The fragment refers to all the contents of an event in the music score or the audio recording with the label $f^{(x)}$. Examples of fragment labels $f^{(x)}$ may be "the Teslim section in the music score of a composition" or "the first 15 seconds of an audio recording."

   A fragment lies in the time-interval $t\big(\bar{f}^{(x)}\big) = \big[t_{ini}\big(\bar{f}^{(x)}\big)\ t_{fin}\big(\bar{f}^{(x)}\big)\big]$ (in seconds). The duration of a fragment $d\big(\bar{f}^{(x)}\big)$ is equal to $\big|t\big(\bar{f}^{(x)}\big)\big| = t_{fin}\big(\bar{f}^{(x)}\big) - t_{ini}\big(\bar{f}^{(x)}\big)$. For a score fragment $\bar{f}^{(b)}$, the duration is equal to the sum of the note durations $\sum\limits_j \bar{n}_j^{(\bar{f}^{(x)})}$, $\bar{n}_j^{(\bar{f}^{(x)})} \in \bar{\mathbf{N}}^{(\bar{f}^{(x)})}$ where $\bar{\mathbf{N}}^{(\bar{f}^{(x)})}$ is the note sequence in the score fragment $(f^{(x)})$.

   The set of fragments $\bar{f}_k^{(x)}$ with the identical label $f^{(x)}$ in a music score or an audio recording (x) is denoted as $\bar{\boldsymbol{F}}^{(x)}(f^{(x)}) = \Big\{\bar{f}_k^{(x)} | f_k^{(x)} = f^{(x)}\Big\}$.

Below, the definitions are extended to the sections in the audio recordings and music scores:

1. We define the *section sequence* in the music score (b) as $\bar{\mathbf{S}}^{(b)} := \Big[\bar{s}_1^{(b)}, \ldots, \bar{s}_{|\bar{\mathbf{S}}^{(b)}|}\Big]$, with each $\bar{s}_j^{(b)}$ consisting of a *section label*, $s_j^{(b)}$ (e.g. "Teslim, Aranağme"), and a note sequence $\bar{\mathbf{N}}^{\left(\bar{s}_j^{(b)}\right)} = \left[\bar{n}_1^{\left(\bar{s}_j^{(b)}\right)},\ \bar{n}_2^{\left(\bar{s}_j^{(b)}\right)}, \ldots\right]$, where $\bar{\mathbf{N}}^{\left(\bar{s}_j^{(b)}\right)}$ is a subsequence of $\bar{\mathbf{N}}^{(b)}$. The note sequence of the sections with the same section label (e.g. repetitive sections) do not have to be identical due to different ending measures, volta brackets etc.

2. The sections in the score form the *score section label sequence*, $\mathbf{S}^{(b)} := \Big[s_1^{(b)}, \ldots, s_{|\mathbf{S}^{(b)}|}^{(b)}\Big]$, where $s_j^{(b)} \in \mathcal{S}_s$ and $j \in$

$\left[1 : |\mathbf{S}^{(b)}|\right]$, with $|\mathbf{S}^{(b)}|$ being the number of sections in a score, repeated sections are counted individually.

3. Let $\mathcal{S}^{(b)} = \left\{ {\mathcal{S}_s}^{(b)}, unrelated \right\}$ denote the *set of section labels*. It consists of a set of symbols ${\mathcal{S}_s}^{(b)} := \left\{ \mathbf{S}^{(b)} \right\}$, which represents all the $|{\mathcal{S}_s}^{(b)}|$ possible distinct section labels in the composition; and an "$unrelated$" section, i.e. a segment with content not related to any structural element of the musical form. The number of unique sections is $\left|\mathcal{S}^{(b)}\right| = |{\mathcal{S}_s}^{(b)}| + 1$.

4. Analogous, for the audio recording (a) we have the (true) *audio section sequence*, $\bar{\mathfrak{S}}^{(a)} = \left[\bar{\mathfrak{s}}_1^{(a)}, \ldots, \bar{\mathfrak{s}}_{|\bar{\mathfrak{S}}^{(a)}|} \right]$. Each element of the sequence, $\bar{\mathfrak{s}}_i^{(a)}$ ($i \in [1 : |\bar{\mathfrak{S}}^{(a)}|]$), has the section label, $\mathfrak{s}_i^{(a)}$, and covers a time interval in the audio, $t\left(\bar{\mathfrak{s}}_i^{(a)}\right)$, i.e. $\bar{\mathfrak{s}}_i^{(a)} = \left\langle \bar{\mathbf{N}}^{\left(\bar{\mathfrak{s}}_i^{(a)}\right)}, \ \mathfrak{s}_i^{(a)}, \ t\left(\bar{\mathfrak{s}}_i^{(a)}\right) \right\rangle$. The time interval is given as $t\left(\bar{\mathfrak{s}}_i^{(a)}\right) = \left[t_{ini}\left(\bar{\mathfrak{s}}_i^{(a)}\right) \ t_{fin}\left(\bar{\mathfrak{s}}_i^{(a)}\right) \right]$, where $t_{ini}\left(\bar{\mathfrak{s}}_1^{(a)}\right) = 0$ sec; $t_{fin}\left(\bar{\mathfrak{s}}_i^{(a)}\right) = t_{ini}\left(\bar{\mathfrak{s}}_{i+1}^{(a)}\right), \forall i \in \left[1 : \ |\bar{\mathfrak{S}}^{(a)}| - 1\right]$; and $t_{fin}\left(\bar{\mathfrak{s}}_{|\bar{\mathfrak{S}}^{(a)}|}^{(a)}\right)$ refers to the end of the audio recording.

5. For the audio recording we have the (true) *audio section label sequence*, $\mathfrak{S}^{(a)} = \left[\mathfrak{s}_1^{(a)}, \mathfrak{s}_2^{(a)}, \ldots \right]$, where $\mathfrak{s}_i^{(a)} \in \mathcal{S}$, $i \in \left[1 : |\mathfrak{S}^{(a)}|\right]$, with $|\mathfrak{S}^{(a)}|$ being the number of sections in the performance, including possibly multiple unrelated sections.

6.2 Melodic Feature Extraction

Score and audio recording are different ways to represent music. To compare these information sources, features that capture the melodic content given in each representation are extracted.

Thoughout the Chapter, predominant melody is typically extracted from the audio recordings (Section 5.2). As mentioned in Section 5.2, either of BOZ-YIN$_f$, SEN-MEL or ATL-MEL is utilized in the experiments. In parallel, synthetic melody is extracted from

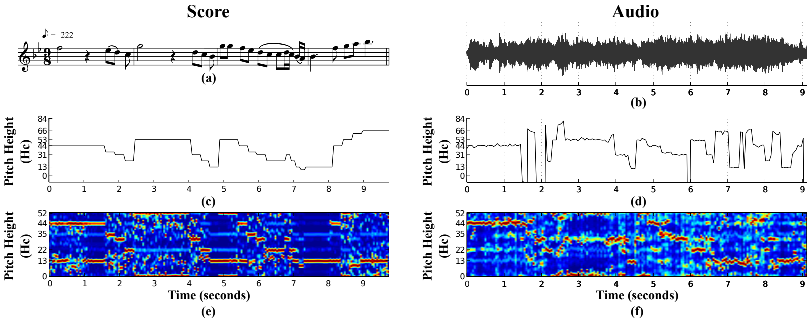


Figure 6.1: Score and audio representations of the first nakarat section of *Gel Güzelim* and the features computed from these representations. a) Score. b) Annotated section in the audio recording. c) Synthetic predominant melody computed from the note symbols and durations. d) predominant melody computed from the audio recording using `SEN-MEL`. The end of the predominant melody has a considerable number of octave errors. e) HPCPs computed from the synthesized MIDI. f) HPCPs computed from the audio recording.

the music scores (Section 4.2.2). The pitch intervals in the synthetic melody are computed either according to the performed tuning (Section 5.9) extracted from the audio recording to be aligned (Appendix A) or according to the AEU theory (Section 6.7.4).

In section linking experiments (Section 6.7.4), the predominant melody (Section 5.2) and the synthetic melody (Section 4.2.2) is compared with the HPCPs (Section 5.3) and the synthetic HPCPs (Section 4.2.3), respectively.

The details of the relevant extraction methods will be given in the respective Sections. Figure 6.1 shows a score excerpt and an audio waveform[1] of the first nakarat section of the composition, *Gel Güzelim*[2] with the features extracted from these sources.

[1] `http://musicbrainz.org/recording/e7be8c2a-3309-4106-93b7-76cd6102a924`

[2] `http://musicbrainz.org/work/9aaf5c0b-4642-40fd-97ba-c861265872ce`

6.3 Fragment Linking

In several tasks such as tonic identification (Section 6.4), tempo estimation (Section 6.5) and composition identification (Figure 6.6), a complete alignment between the audio recording and music score is not necessary. Moreover, structural differences between the music score and audio performance of OTMM (Section 2.1) have to be taken care of for complete alignment. This problem is approached from a bottom-up scheme by first aligning the sections in the music score separately and then estimating the global alignment from the individual paths (Section 6.7). Therefore, obtaining partial alignments between the audio recording and the score fragment(s) is the fundamental step in all audio-score alignment tasks described hereafter.

Fragment linking is defined as "marking the fragments, in an audio recording at which a fragment selected from a music score are performed." Given the fragment $\bar{f}^{(b)}$ selected from the music score (b) with the label $f^{(b)}$, the set of fragments in the audio recording (a) with the same label is denoted as $\bar{\boldsymbol{F}}^{(a)}(f^{(b)}) = \{\bar{f}_1^{(a)}, \bar{f}_2^{(a)}, \dots\}$, where $\bar{f}_m^{(a)} = f^{(b)}$ and $m \in \left[1 : |\bar{\boldsymbol{F}}^{(a)}(f^{(b)})|\right]$ in the time interval $t\left(\bar{f}_m^{(a)}\right) = \left[t_{ini}\left(\bar{f}_m^{(a)}\right)\ t_{fin}\left(\bar{f}_m^{(a)}\right)\right]$.

This process forms a link, $\pi(\bar{f}_m^{(a)}, \bar{f}^{(b)})$, between the score fragment $\bar{f}^{(b)}$ and each audio fragment $\bar{f}_m^{(a)}$ such that $f_m^{(a)} = f^{(b)}$, $\forall \bar{f}_m^{(a)}$, and the start t_{ini} and the end t_{fin} of the linked fragments are mapped to each other. Linking the audio and score fragments also imply relating the relevant attributes of each source (if available), e.g. the tonic symbol $\kappa^{(b)}$ of the score and the tonic frequency $\kappa^{(a)}$ of the audio recording. Similarly, additional useful information may be obtained from the linking process, e.g. the intra-alignment path, $\boldsymbol{\varpi}(\bar{f}_m^{(a)}, \bar{f}^{(b)})$ between the fragments $\bar{f}_m^{(a)}$ and $\bar{f}^{(b)}$, which can be used later in note-level alignment (Section 6.8). The set of links between the audio recording (a) and the music score (b) with the fragment label (x) is denoted as $\Pi(a, b, x) = \left\{\pi(\bar{f}_m^{(a)}, \bar{f}^{(b)}),\ f_m^{(a)} = f^{(b)} = x\right\}$.

Two different methods are used for fragment linking: 1) Hough transform (Section 6.3.1), and 2) Subsequence dynamic time warping (Section 6.3.2). Throughout this Section, it is assumed that the tonic pitch/pitch-class of the audio recording, $\kappa^{(a)}$, is already identified. The tonic identification step will be explained in Section 6.4.

6.3.1 Hough Transform

Given an audio recording (a) of a composition and a fragment $\bar{f}^{(b)}$ selected from the music score (b) of the same composition, we first extract either the predominant melody or the HPCPs from the audio recording. The audio feature is then normalized with respect to the performance tonic, $\kappa^{(a)}$, and a normalized predominant melody, $\hat{\boldsymbol{\varrho}}^{\kappa^{(a)},(a)}$, or HPCPs, $\hat{\boldsymbol{\Gamma}}^{\kappa^{(a)},(a)}$, are obtained. In parallel, a synthetic melody, $\hat{\boldsymbol{\Psi}}^{(\bar{f}^{(b)})}$, or synthetic HPCPs, $\hat{\boldsymbol{\Omega}}^{(\bar{f}^{(b)})}$, are computed. To compare the audio recording with each section in the score, we compute a distance matrix between the score feature, $\hat{\boldsymbol{\Psi}}^{(\bar{f}^{(b)})}$ or $\hat{\boldsymbol{\Omega}}^{(\bar{f}^{(b)})}$, of the fragment $(\bar{f}^{(b)})$ and the audio feature, $\hat{\boldsymbol{\varrho}}^{\kappa^{(a)},(a)}$ or $\hat{\boldsymbol{\Gamma}}^{\kappa^{(a)},(a)}$, of the whole recording.

If predominant melodies are chosen as the features, the distance matrix, $\boldsymbol{D}^{\kappa^{(a)},(a,\bar{f}^{(b)})}$, between the normalized audio predominant melody, $\hat{\boldsymbol{\varrho}}^{\kappa^{(a)},(a)}$, and the synthetic predominant melody, $\hat{\boldsymbol{\Psi}}^{(\bar{f}^{(b)})}$, of a score fragment $(\bar{f}^{(b)})$ is obtained by computing the pairwise smallest Hc distance between each point of the features using Equation 5.3. To recapitulate, each element $\boldsymbol{D}_{ij}^{\kappa^{(a)},(a,\bar{f}^{(b)})}$ in the distance matrix $\boldsymbol{D}^{\kappa^{(a)},(a,\bar{f}^{(b)})}$ is computed as:

$$\boldsymbol{D}_{ij}^{\kappa^{(a)},(a,\bar{f}^{(b)})} = \blacktriangle\left(\hat{\psi}_i^{(\bar{f}^{(b)})}, \hat{\rho}_j^{\kappa^{(a)},(a)}\right) \tag{6.1}$$

where $\hat{\psi}_i^{(\bar{f}^{(b)})}$ is the i^{th} sample of the synthetic melody $\hat{\boldsymbol{\Psi}}^{(\bar{f}^{(b)})}$ computed from the score fragment $\bar{f}^{(b)}$ (Section 4.2.2), $\hat{\rho}_j^{\kappa^{(a)},(a)}$ is the j^{th} sample of the predominant melody $\hat{\boldsymbol{\varrho}}^{\kappa^{(a)},(a)}$ normalized with respect to the tonic $\kappa^{(a)}$ of the audio recording (a) (Section 5.2), and $\blacktriangle(x, y)$ denotes the shortest octave-wrapped distance between the pitch values x and y (Section 5.4). The first element in the superscript, $\kappa^{(a)}$, indicates that the predominant melody, $\hat{\boldsymbol{\varrho}}^{\kappa^{(a)},(a)}$, extracted from (a) is normalized to cent-scale by taking the tonic $\kappa^{(a)}$ as the reference. $\boldsymbol{D}^{\kappa^{(a)},(a,\bar{f}^{(b)})}$ is a $\left|\hat{\boldsymbol{\varrho}}^{\kappa^{(a)},(a)}\right|$ x $\left|\hat{\boldsymbol{\Psi}}^{(\bar{f}^{(b)})}\right|$ matrix. Figure 6.2e shows a distance matrix computed for the Teslim section between an audio performance[3] and **SymbTr**-score of *Şedaraban Sazsemaisi*.[4]

If HPCPs are chosen as the input features, the distance matrix, $\boldsymbol{D}^{\kappa^{(a)},(a,\bar{f}^{(b)})}$, between the normalized audio HPCPs, $\hat{\mathbf{\Gamma}}^{\kappa^{(a)},(a)}$, and the synthetic HPCPs, $\hat{\mathbf{\Omega}}^{(\bar{f}^{(b)})}$, of a score fragment, $(\bar{f}^{(b)})$, is obtained by taking the *cosine* distance between each HPCP frame. Cosine distance is a common feature used for comparing chroma features (Paulus et al., 2010) computed as:

$$\boldsymbol{D}_{ij}^{\kappa^{(a)},(a,\bar{f}^{(b)})} = 1 - \frac{\sum\limits_{k=1}^{n_{\text{HPCP}}} \hat{\omega}_{ik}^{(\bar{f}^{(b)})} \ \hat{\gamma}_{jk}^{\kappa^{(a)},(a)}}{\sqrt{\left(\sum\limits_{k=1}^{n_{\text{HPCP}}} \left(\hat{\omega}_{ik}^{(\bar{f}^{(b)})}\right)^2\right) \cdot \left(\sum\limits_{k=1}^{n_{\text{HPCP}}} \left(\hat{\gamma}_{jk}^{\kappa^{(a)},(a)}\right)^2\right)}},$$

$$1 \leq i \leq |\hat{\mathbf{\Omega}}^{(\bar{f}^{(b)})}| \text{ and } 1 \leq j \leq |\hat{\mathbf{\Gamma}}^{\kappa^{(a)},(a)}| \quad (6.2)$$

where $\hat{\omega}_{ik}^{(\bar{f}^{(b)})}$ is the k^{th} bin of the i^{th} frame of the HPCPs (of $\left|\hat{\mathbf{\Omega}}^{(\bar{f}^{(b)})}\right|$ frames) of a fragment $(\bar{f}^{(b)})$, $\hat{\gamma}_{jk}^{\kappa^{(a)},(a)}$ is the k^{th} bin of the j^{th} frame of the HPCPs (of $\left|\hat{\mathbf{\Gamma}}^{\kappa^{(a)},(a)}\right|$ frames) extracted from the audio recording (a) and n_{HPCP} denotes the number of bins per frame chosen for the HPCP computation. Cosine distance between two HPCP frames of length n_{HPCP} can be interpreted as 1 minus the dot product of the two frames, normalized to unit length on an n_{HPCP}-dimensional Euclidean space. The outcome is bounded to the interval $[0\ 1]$ for non-negative inputs, 0 denoting the "closest," which makes it possible to compare the relative distance between the frames of HPCPs that have unitless, non-negative values.

If the score fragment is performed in the audio, the distance matrix shows blob(s) in a diagonal trajectory formed by low distance

[3] http://musicbrainz.org/recording/efae832f-1b2c-4e3f-b7e6-62e08353b9b4

[4] http://musicbrainz.org/work/1eb2ca1e-249b-424c-9ff5-0e1561590257

values, which hint the location(s) of the score fragment in the audio (Figure 6.2e). The values of the points forming the blobs may be substantially greater than zero in practice, making it harder to distinguish the blobs from the background. Therefore, binary thresholding is applied to the distance matrix before applying Hough transform to emphasize the diagonal blobs. A binary similarity matrix $\boldsymbol{B}^{\kappa^{(a)},(a,\bar{f}^{(b)})}$ is obtained as:

$$\boldsymbol{B}_{ij}^{\kappa^{(a)},(a,\bar{f}^{(b)})} = \begin{cases} 1, & \boldsymbol{D}_{ij}^{\kappa^{(a)},(a,\bar{f}^{(b)})} \leq \beta(\boldsymbol{B}) \\ 0, & \boldsymbol{D}_{ij}^{\kappa^{(a)},(a,\bar{f}^{(b)})} > \beta(\boldsymbol{B}) \end{cases} \tag{6.3}$$

where $\beta(\boldsymbol{B})$ is the binarization threshold. The binary similarity matrix $\boldsymbol{B}_{ij}^{\kappa^{(a)},(a,\bar{f}^{(b)})}$ of a fragment $\bar{f}^{(b)}$ shows which samples between the score feature and the audio feature are similar enough to each other to be deemed as the same note (Figure 6.2f). For melody, the binarization threshold correspond to the minimum value of shortest octave wrapped distance in cents. For HPCPs, the binarization threshold is unitless in the interval $[0:1]$. Effect of the binarization threshold value is investigated in the section linking experiments (Section 6.7.4). In the preliminary section linking experiments, we additionally used a number of structural morphological operations (Serra, 1983; Ballard, 1981) to emphasize the blobs. They are explained in Appendix A separately for the sake of brevity.

As can be seen in Figure 6.2f, these blobs can be approximated as line segments. To detect these segments, a common line detection method called Hough transform is applied to the binary similarity matrix (Figure 6.2g) (Ballard, 1981). Hough transform has also been previously used in musical tasks such as locating the formant trajectories of drum beats (Townsend & Sandler, 1993) and detecting repetitive structures in an audio recording for thumbnailing (Aucouturier & Sandler, 2002).

The angle $\theta\left(\bar{f}_m^{(a)}, \bar{f}^{(b)}\right)$ of a detected diagonal line segment is related to the tempo of the performed audio fragment $\tau^{\left(\bar{f}_m^{(a)}\right)}$ and the tempo of the respective score fragment $\tau^{\left(\bar{f}^{(b)}\right)}$. We define the *relative tempo* for each candidate $\hat{\tau}^{\left(\bar{f}_m^{(a)}\right)}$ as:

$$\hat{\tau}^{\left(\bar{f}_m^{(a)}\right)} = \tan\Big(\theta\big(\bar{f}_m^{(a)}, \bar{f}^{(b)}\big)\Big) = \frac{d\left(\bar{f}^{(b)}\right)}{d\left(\bar{f}_m^{(a)}\right)} \approx \frac{\tau^{\left(\bar{f}_m^{(a)}\right)}}{\tau^{\left(\bar{f}^{(b)}\right)}}, f_k^{(a)} = f^{(b)} \tag{6.4}$$

where $d\left(\bar{f}^{(b)}\right)$ is the duration of the fragment given in the score, $d\left(\bar{f}_m^{(a)}\right) = \left|t\left(\bar{f}_m^{(a)}\right)\right| = t_{fin}\left(\bar{f}_m^{(a)}\right) - t_{ini}\left(\bar{f}_m^{(a)}\right)$ is the duration of the candidate audio fragment $\bar{f}_m^{(a)}$ and $\theta\left(\bar{f}_m^{(a)}, \bar{f}^{(b)}\right)$ is the angle of the line segment associated with the candidate fragment link. Provided that there are no phrase repetitions, omissions or extreme tempo changes inside the performed fragment, relative tempo approximately indicates the amount of deviation from the tempo given in the score. If the tempo of the performance is exactly the same with the tempo, the angle of the diagonal line segment is $45°$.

The relative tempo $\hat{\tau}^{\left(\bar{f}_m^{(a)}\right)}$ of a fragment candidate is constrained between 0.5 and 2. This limits the angles searched in Hough transform to an interval $[\theta_{min},\ \theta_{max}]$:

$$[\theta_{min},\ \theta_{max}] = \begin{cases} \theta_{min} = \text{arctan}(0.5) \approx 27° \\ \theta_{max} = \text{arctan}(2) \approx 63° \end{cases} \tag{6.5}$$

The step size of the angles between θ_{min} and θ_{max} is set to 1 degree. Then, the peaks are selected from the obtained transformation matrix. These peaks indicate the angle and the distance to the origin of the most prominent line segments (Duda & Hart, 1972). Considering the maximum number of repetitions in the peşrev, sazsemaisi and şarkı forms (Section 2.1) plus a tolerance of 50%, we pick the 12 peaks emitting the highest similarity. Next, the line segments (i.e. the linear alignment paths, $\varpi\left(\bar{f}_m^{(a)}, \bar{f}^{(b)}\right)$ between the score fragment $\bar{f}^{(b)}$ and the estimated audio fragments $\bar{f}_m^{(a)}$ are computed from this set of peaks such that the line segment covers the entire duration of the section given in the score (Figure 6.2g).

6.3.2 Subsequence Dynamic Time Warping

Dynamic programming and more specifically DTW are the state-of-the-art methodologies for many relevant tasks such as cover song

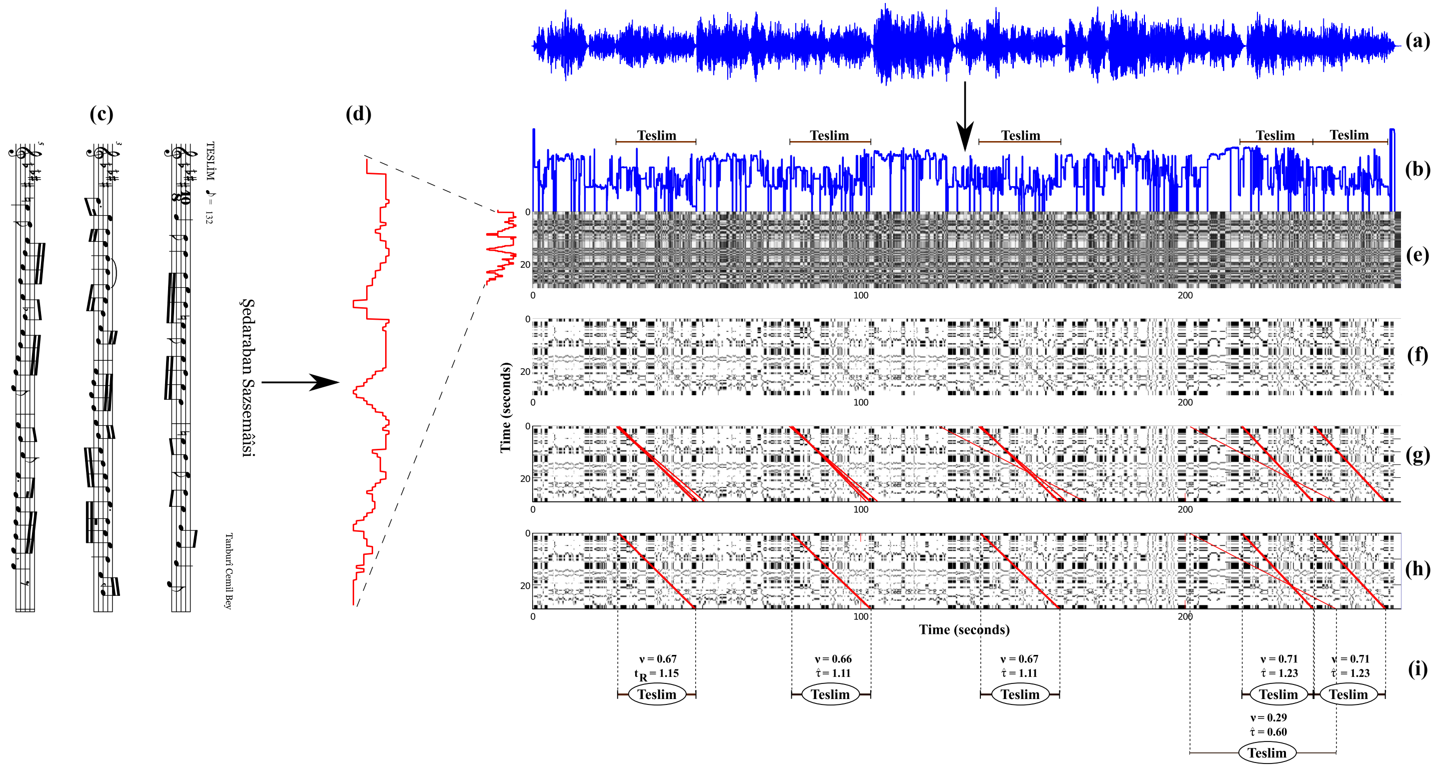


Figure 6.2: Steps in linking the Teslim section of the *Şedaraban Sazsemaisi* and an audio recording of the composition using Hough transform for alignment, and predominant melody and synthetic melody as the input features. **a)** Audio waveform. **b)** Predominant melody extracted from the audio recording. Annotated Teslims are shown over the predominant melody. **c)** Teslim section of the *Şedaraban Sazsemaisi* **d)** Synthetic pitch computed from the Teslim section. **e)** Distance matrix between the predominant melody and the synthetic melody. White indicates the closest distance (0 Hc). **f)** Image binarization on distance matrix. White and black represent zero (dissimilar) and one (similar) respectively. **g)** Line detection using Hough transform. **h)** Elimination of duplicates. **i)** Teslim candidates. The numerical values "ν" and "$\hat{\tau}$" indicate the similarity and the relative tempo of the candidate respectively.

identification (Serrà et al., 2009; Ellis & Poliner, 2007) and audio score alignment (Müller & Appelt, 2008; Niedermayer, 2012). Unlike Hough transform, DTW is robust to changes in tempo and musical insertions, deletions and repetitions. However, it can be prone to pathological warpings.

Subsequence dynamic time warping is a typical variant of DTW used, when one of the time series is a subsequence of the other (Anguera & Ferrarons, 2013; Müller, 2007). In this variant the paths are allowed to start/end within the target. We refer the readers to (Müller, 2007, Chapter 4) for a thorough explanation of DTW and its variants.

We use SDTW in composition identification (Section 6.6) and note-level alignment (Section 6.8). Selecting the predominant melody as the input feature, we first compute an element $\boldsymbol{A}_{ij}^{\kappa^{(a)},(a,\bar{f}^{(b)})}$ in the accumulated cost matrix $\boldsymbol{A}^{\kappa^{(a)},(a,\bar{f}^{(b)})}$ recursively as:

$$\boldsymbol{A}_{ij}^{\kappa^{(a)},(a,\bar{f}^{(b)})} = \begin{cases} 0, & i=0 \\ +\infty, & j=0 \\ \boldsymbol{D}_{ij}^{\kappa^{(a)},(a,\bar{f}^{(b)})} + min \begin{cases} \boldsymbol{A}_{i-1\,j-1}^{\kappa^{(a)},(a,\bar{f}^{(b)})} & \\ \boldsymbol{A}_{i-2\,j-1}^{\kappa^{(a)},(a,\bar{f}^{(b)})}, & i>1 \\ \boldsymbol{A}_{i-1\,j-2}^{\kappa^{(a)},(a,\bar{f}^{(b)})}, & j>1 \end{cases} & i,j\neq 0 \end{cases} \tag{6.6}$$

As seen above, we select the step size condition as $\{(2,1), (1,1),(1,2)\}$. Analogous to the angle restriction in Hough transform (Section 6.3.1), this step size ensures that the intra-tempo variations in any path will stay between half and double the nominal tempo indicated in the score. We use the local distance $\boldsymbol{D}_{ij}^{\kappa^{(a)},(a,\bar{f}^{(b)})}$ to calculate the accumulated cost matrix. Also, notice that the accumulated cost matrix is extended with a zeroth row and column, initialized to enable subsequence matching.

We then back-track the path $\boldsymbol{\varpi}\left(\bar{f}_m^{(a)},\bar{f}^{(b)}\right)$ ending at $\text{argmin}_{(i)}$ $\boldsymbol{A}_{i\left|\hat{\boldsymbol{\Psi}}^{f^{(b)}}\right|}^{\kappa^{(a)},(a,\bar{f}^{(b)})}$ (remember that $\left|\hat{\boldsymbol{\Psi}}^{f^{(b)}}\right|$ is the length of the synthetic melody), which emits the lowest accumulated cost (Müller, 2007, Chapter 4).

In the proposed composition identification and note-level alignment methodologies, we align a single fragment as will be explained in Section 6.6 and Section 6.8. If multiple estimations are required (e.g. in the audio-score alignment methodology adapted for Carnatic music as explained in Appendix B.1), one can use ISDTW. In ISDTW, a path is back-tracked as explained in the case of SDTW. Next, the vicinity of the path (e.g. 10% of the alignment path length $\left|\boldsymbol{\varpi}\left(\bar{f}_m^{(a)}, \bar{f}^{(b)}\right)\right|$ on the audio axis) is "blacklisted" by assigning $+\infty$ to the respective values in the accumulated cost matrix. Then, a new path is back-tracked by referring to the minima of the row $\boldsymbol{A}^{\kappa^{(a)},(a,\bar{f}^{(b)})}_{i\left|\hat{\boldsymbol{\Psi}}^{f^{(b)}}\right|}$ in the recomputed accumulated cost matrix. By repeating this process at most 12 times (analogous to peak picking in Hough transform),[5] the estimated audio fragments $[\bar{f}_1^{(a)}, \bar{f}_2^{(a)}, \ldots]$ are obtained. The tempo $\tau^{\left(\bar{f}_m^{(a)}\right)}$ and the relative tempo $\hat{\tau}^{\left(\bar{f}_m^{(a)}\right)}$ are estimated from the durations covered by path on the music score and the audio recording using Equation 6.4.

6.3.3 Similarity Computation

Using either Hough transform or SDTW, we obtain a set of alignment paths $\left\{\boldsymbol{\varpi}\left(\bar{f}_1^{(a)}, \bar{f}^{(b)}\right), \boldsymbol{\varpi}\left(\bar{f}_2^{(a)}, \bar{f}^{(b)}\right), \ \ldots\right\}$. Each alignment path $\boldsymbol{\varpi}\left(\bar{f}_m^{(a)}, \bar{f}^{(b)}\right)$ between the audio fragment $\bar{f}_m^{(a)}$ and the score fragment $\bar{f}^{(b)}$ is denoted as:

$$\boldsymbol{\varpi}\big(\bar{f}_m^{(a)}, \bar{f}^{(b)}\big) = \left[\varpi_1\big(\bar{f}_m^{(a)}, \bar{f}^{(b)}\big), \ldots, \varpi_{\left|\boldsymbol{\varpi}\left(\bar{f}_m^{(a)}, \bar{f}^{(b)}\right)\right|}\big(\bar{f}_m^{(a)}, \bar{f}^{(b)}\big)\right] \tag{6.7}$$

where $\varpi_l\left(\bar{f}_m^{(a)}, \bar{f}^{(b)}\right) = \left(r_l\big(\bar{f}_m^{(a)}\big), q_l\big(\bar{f}^{(b)}\big)\right)$. Here $r_l\left(\bar{f}_m^{(a)}\right)$ and $q_l\big(\bar{f}^{(b)}\big)$ refer to the sample indices in the feature extracted from the audio recording (i.e. the $r_l\big(\bar{f}_m^{(a)}\big)^{\text{th}}$ pitch $\hat{\rho}^{(a)}_{r_l\left(\bar{f}_m^{(a)}\right)}$ of the predominant melody $\hat{\boldsymbol{\varrho}}^{(a)}$ or the $r_l\big(\bar{f}_m^{(a)}\big)^{\text{th}}$ frame $\hat{\gamma}^{(a)}_{r_l\left(\bar{f}_m^{(a)}\right)}$ of the HPCPs $\hat{\boldsymbol{\Gamma}}^{(a)}$) and from the music score (i.e. the $q_l\big(\bar{f}_m^{(a)}\big)^{\text{th}}$ pitch $\hat{\psi}^{(b)}_{q_l\left(\bar{f}_m^{(a)}\right)}$

[5] If most of the accumulated cost matrix is blacklisted such that no other path can be backtracked, the process will return less than 12 paths.

of the synthetic melody $\hat{\mathbf{\Psi}}^{(b)}$ or the $q_l\left(\bar{f}_m^{(a)}\right)^{\text{th}}$ frame $\hat{\omega}^{(b)}_{q_l\left(\bar{f}_m^{(a)}\right)}$ of the synthetic HPCPs $\hat{\mathbf{\Omega}}^{(b)}$), respectively. The index l is an element of $\left[1 : \left|\varpi\left(\bar{f}_m^{(a)}, \bar{f}^{(b)}\right)\right|\right]$, where $\left|\varpi\left(\bar{f}_m^{(a)}, \bar{f}^{(b)}\right)\right|$ is the length of the path $\varpi\left(\bar{f}_m^{(a)}, \bar{f}^{(b)}\right)$. We compute a similarity, $\nu(\bar{f}_m^{(a)}, \bar{f}^{(b)}) \in [0:1]$, between the score fragment $\bar{f}^{(b)}$ and the audio fragment $\bar{f}_m^{(a)}$ by:

$$\nu(\bar{f}_m^{(a)}, \bar{f}^{(b)}) = \frac{\sum_l B\left(r_l(\bar{f}_m^{(a)}), q_l(\bar{f}^{(b)})\right)}{\left|\varpi\left(\bar{f}_m^{(a)}, \bar{f}^{(b)}\right)\right|} \tag{6.8}$$

6.3.4 Duplicate Link Removal

If two or more links have their borders in the same vicinity (±6 seconds), they are treated as duplicates. This occurs frequently using Hough transform since the line segments in the binary matrix are actually blobs. Hence, there might be line segments with slightly different parameters, effectively estimating the same candidate. Among the duplicates, only the one with the highest similarity is kept (Figure 6.2h).

Finally, the regions covered by the remaining estimations are chosen as the candidate audio fragments, $\bar{f}_m^{(a)}$, with the time intervals, $t(\bar{f}_m^{(a)}) = \left[t_{ini}(\bar{f}_m^{(a)})\ t_{fin}(\bar{f}_m^{(a)})\right]$, and the tempi, $\tau^{\left(\bar{f}_m^{(a)}\right)}$.[6] For each fragment, $\bar{f}_m^{(a)}$, a link, $\pi\left(\bar{f}_m^{(a)}, \bar{f}^{(b)}\right)$ is formed (such that $f_m^{(a)} = f^{(b)}$) with the similarity, $\nu(\bar{f}_m^{(a)}, \bar{f}^{(b)})$, and the alignment path, $\varpi\left(\bar{f}_m^{(a)}, \bar{f}^{(b)}\right)$ (Figure 6.2i).

6.4 Score-Informed Tonic Identification

In Section 5.7, a tonic identification methodology, which is based on matching the distribution extracted from an audio recording with unknown tonic with template distributions obtained from a set of

[6] and therefore the relative tempi $\hat{\tau}^{\left(\bar{f}_m^{(a)}\right)}$

training audio recordings, is described. The disadvantage of using audio recordings for template computation is the necessity of adequate amount of training data. Moreover, the quality of the data has to be maintained so that the intervallic properties are represented well. Even so, a test distribution can substantially differ from the corresponding template. A common confusion is the estimation of another pitch (or pitch class) when its occurrence is comparable to the occurrence of the tonic. Moreover, in cases when an audio recording includes unrelated musical content in addition to a performed piece, e.g. improvisations or performances of other pieces in different modal structures, the resultant distribution would be a mixture of the distributions of these distinct musical events. This might cause substantial confusions. This problem motivates the replacement of audio recordings with a more "definitive" information source in the template training step. If available, scores can be good sources, since they provide an easily accessible symbolic description of many relevant musical components.

The remainder of the Section is organized as follows: Section 6.4.1 explain the proposed methodologies. Section 6.4.2 explains the dataset and the experimental setup to test the methodologies and provides the results. Section 6.4.5 wraps up the Section with a discussion and conclusion.

6.4.1 Methodologies

In (Şentürk et al., 2013), two methodologies, which identify the tonic of a performed piece, are presented. In this task, the audio recording and the score are already known to be related with the same work (composition) and the music scores (**SymbTr**-txt) include the makam of the piece, the boundaries of the structural elements and the sequence of these elements. We use makam music knowledge and the findings from previous research (Gedik & Bozkurt, 2010; Chordia & Şentürk, 2013; Şentürk, Holzapfel, & Serra, 2014; Karakurt et al., 2016) to specialise both the methodologies for the melodic aspects of OTMM. Both methods extract predominant melody from the audio recording (Section 5.2). Then a PCD is computed from the audio predominant melody (Section 5.5) and stable pitch classes are selected as the tonic candidates (Section 5.6). Adapted from (Gedik & Bozkurt, 2010; Chordia & Şentürk, 2013),

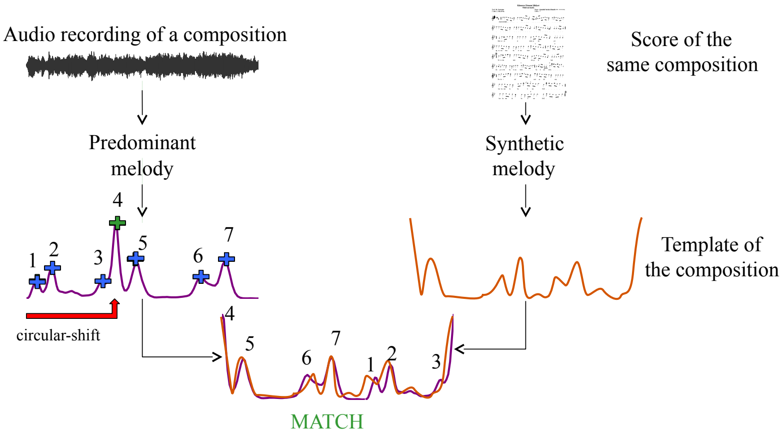

Audio recording of a composition
Score of the same composition
Predominant melody
Synthetic melody
1
2
3
4
5
6
7
Template of the composition
circular-shift
4
5
6
7
1
2
3
MATCH

Figure 6.3: Tonic identification by music score template PCD matching

the first method applies circular shifting to the audio PCD according to tonic candidates. Each shift is then compared with a score PCD computed from the monophonic melody in the score. The second method normalises the predominant melody with respect to each tonic candidate. Next, it attempts to link melodic fragments in the score with the respective time intervals in the audio recordings by using the fragment linking approach explained in (Şentürk, Holzapfel, & Serra, 2014) (Section 6.3).

Note that a recording may contain performances of multiple compositions, which may have different performance tonics.[7] We can estimate the tonic $\kappa^{(a,b_i)}$ of the sub-performance in the audio recording (a), in which the composition (b_i) is performed. This case is handled within the combined joint analysis procedure described in Section 6.12. For the sake of brevity, the score-informed tonic identification process between a single audio recording (a) and music score (b) is described in this Section. The score relation, (b_i), in the the tonic symbol, $\kappa^{(a,b_i)}$, is omitted and the symbol is denoted as $\kappa^{(a)}$.

Tonic Identification by Score Template Distribution Matching

The tonic identification method by score template distribution matching (Şentürk et al., 2013) ($\texttt{SEN-TON}_{PCD}$) first generates a synthetic melody $\hat{\boldsymbol{\Psi}}^{(b)}$ from the complete music score (b) (Section 4.2.2) by referring to the intervals defined by the AEU theory. Next a score PCD, $\hat{\boldsymbol{H}}_{PC}{}^{(b)}$, is computed from the synthetic melody.

In parallel, a predominant melody, $\boldsymbol{\varrho}^{(a)}$, is extracted from the audio recording (a) using `SEN-MEL` (Section 5.2). Then a PCD, $\boldsymbol{H}_{PC}^{(a)}$, is computed from the audio predominant melody (Section 5.5). The first bin of the PCD is initialised to the dummy value, 440.0 Hz, as described in Section 5.5. The bin size, $b\left(\hat{\boldsymbol{H}}\right)$, of the PCD is selected as 7.5 cents and the kernel width, $\sigma\left(\hat{\boldsymbol{H}}\right)$, is set to 15 cents ($\approx \frac{2}{3}$ Hc) empirically, so that an observation practically contributes within an interval of 4Hc (slightly smaller than a semitone). Next, the stable pitch classes, $\boldsymbol{\Phi}_{PC}^{(a)}$, which are extracted from the peaks of the audio PCD (Section 5.6), are selected as the tonic candidates.

The audio PCD is circularly shifted according to its stable pitch classes, $\phi_k^{(a)} \in \boldsymbol{\Phi}_{PC}^{(a)}$, one by one and $\hat{\boldsymbol{H}}_{PC}{}^{\phi_k^{(a)},(a)}$s are obtained. These shifted distributions are compared to the score PCD, $\hat{\boldsymbol{H}}_{PC}{}^{(b)}$, using Bhattacharyya distance (Equation 5.10). The tonic candidate $\phi_k^{(a)}$ used to shift the audio PCD, which results in the minimum distance to score PCD, is selected as the tonic pitch class $\kappa^{(a)}$ (Figure 6.3).

The tonic identification methods using audio template distribution matching (Gedik & Bozkurt, 2010; Chordia & Şentürk, 2013; Karakurt et al., 2016) are based on the similarity of the distributions belonging to the same makam. On the other hand, $\texttt{SEN-TON}_{PCD}$ is based on the similarity of the distributions belonging to the same composition, which is expected to score higher due to less variability in the relative occurence of the pitch classes in the PCDs.

[7] e.g. http://musicbrainz.org/recording/37dd6a6a-4c19-4a86-886a-882840d59518

Tonic Identification by Fragment Linking

Using score PCD templates, we can only take an advantage of the interval and some limited intonation information. Nevertheless, scores also include note sequence information. In the tonic identification method by fragment linking (Şentürk et al., 2013) (`SEN-TON`$_{link}$), we attempt to link a melodic fragment from the score with the audio recording by extending the fragment linking procedure presented in (Şentürk, Holzapfel, & Serra, 2014) (Section 6.3).

`SEN-TON`$_{link}$ uses the note sequence information given in the score. From the score (b), `SEN-TON`$_{link}$ only extracts the synthetic melody $\hat{\boldsymbol{\Psi}}^{\bar{f}^{(b)}}$ of a fragment $\bar{f}^{(b)}$ selected from the score. The fragment may be sliced from the start or the repetitive section (e.g. Teslim, Nakarat) of the score, with a certain duration. Similar to `SEN-TON`$_{PCD}$, `SEN-TON`$_{link}$ computes a predominant melody $\boldsymbol{\varrho}^{(a)}$, audio PCD $\boldsymbol{H}_{PC}^{(a)}$ and the stable pitch classes $\boldsymbol{\Phi}_{PC}^{(a)}$ from the audio recording (a).

The method then obtains a normalized audio predominant melody $\hat{\boldsymbol{\varrho}}^{\phi_k^{(a)},(a)}$ with respect to each stable pitch class $\phi_k^{(a)}$ and attempts to link the fragment in the score with its respective locations $t^{\phi_k^{(a)}}(\bar{f}_m^{(a)})$[8] in the audio recording (Section 6.3). In the experiments explained in Sections 6.4.2, a binary similarity matrix $\boldsymbol{B}^{\phi_k^{(a)},(a,\bar{f}^{(b)})}$ is computed by taking the binarization threshold $\beta(\boldsymbol{B})$ as 3 Hc (Equation 6.3), an optimal for OTMM as will be discussed in Section 6.7.4. Then, Hough transform is applied to the binary similarity matrix to detect the diagonal line segments as described in Section 6.3.1. In the experiments explained in Sections 6.6.3, `SDTW` is compared with Hough transform. There, an accumulated cost matrix $\boldsymbol{A}^{\phi_k^{(a)},(a,\bar{f}^{(b)})}$ is computed using `SDTW` and the path emitting the lowest similarity is backtracked (Section 6.3.2).

We obtain a set of links $\Pi^{\phi_k^{(a)}}(a, b, f^{(b)}) = \left\{\pi^{\phi_k^{(a)}}\left(\bar{f}_1^{(a)}, \bar{f}^{(b)}\right), \pi^{\phi_k^{(a)}}\left(\bar{f}_2^{(a)}, \bar{f}^{(b)}\right), \ldots\right\}$ such that $f^{(b)} = \bar{f}_m^{(a)}$ for each tonic candidate $\phi_k^{(a)}$. The similarity of a link $\pi^{\phi_k^{(a)}}\left(\bar{f}_m^{(a)}, \bar{f}^{(b)}\right)$ is denoted as

[8] Notice that the time interval, links and similarity values are denoted with an additional superscipt in this Section, indicating that the estimated audio fragments $\bar{f}_m^{(a)}$ are dependent on the tonic candiate $\phi_k^{(a)}$.

$\nu^{\phi_k^{(a)}}\left(\bar{f}_m^{(a)}, \bar{f}^{(b)}\right)$.

The similarity of each link may be summarized to obtain a weight for each tonic candidate. The accumulated weight is given as:

$$\nu\left(\phi_k^{(a)}\right) = \sqrt[3]{\sum_{m}^{\left|\Pi^{\phi_k^{(a)}}(a,b,f^{(b)})\right|} \nu^{\phi_k^{(a)}}\left(\bar{f}_m^{(a)}, \bar{f}^{(b)}\right)^3} \tag{6.9}$$

Equation 6.9 ensures that (possibly erroneous) links with low similarities are suppressed with respect to the links with high similarities. The tonic is estimated as the pitch class $\phi_k^{(a)}$, which has the highest accumulated weight $\nu\left(\phi_k^{(a)}\right)$, i.e. $\kappa^{(a)} = \underset{\phi_k^{(a)} \in \mathbf{\Phi}_{PC}^{(a)}}{\text{argmax}}\, \nu\left(\phi_k^{(a)}\right)$.

6.4.2 Experiments

We test the methodologies explained in Section 6.4.1 on **OTMM-section-linking**. In SEN-TON$_{link}$, the fragment is selected as the whole repetitive section and synthesized by referring to the theoretical intervals defined by the AEU theory. We compare the estimated tonic $\kappa^{(a)}$ from each algorithm with the manually annotated tonic $\mathfrak{t}^{(a)}$ using Equation 5.3. If the shortest octave-wrapped distance between the estimated and the annotated tonic are less than 1 Hc, the estimation is marked as *True*. Then, the tonic identification accuracy is computed by dividing the number of *True* identifications to the total number of identifications.[9]

6.4.3 Dataset

To test the methodologies, we use the audio recordings in **OTMM-section-linking** dataset. **OTMM-section-linking** consists of 116 audio recordings of 24 peşrevs, 84 audio recordings of 19 sazsemaisis, and 57 audio recordings of 14 şarkıs (257 audio recordings of 57 compositions in total). The compositions are taken from the classical repertoire, in which the makam and the karar note are clearly defined in music theory. The audio recordings and music scores are

[9] The results are available at `http://compmusic.upf.edu/node/164`.

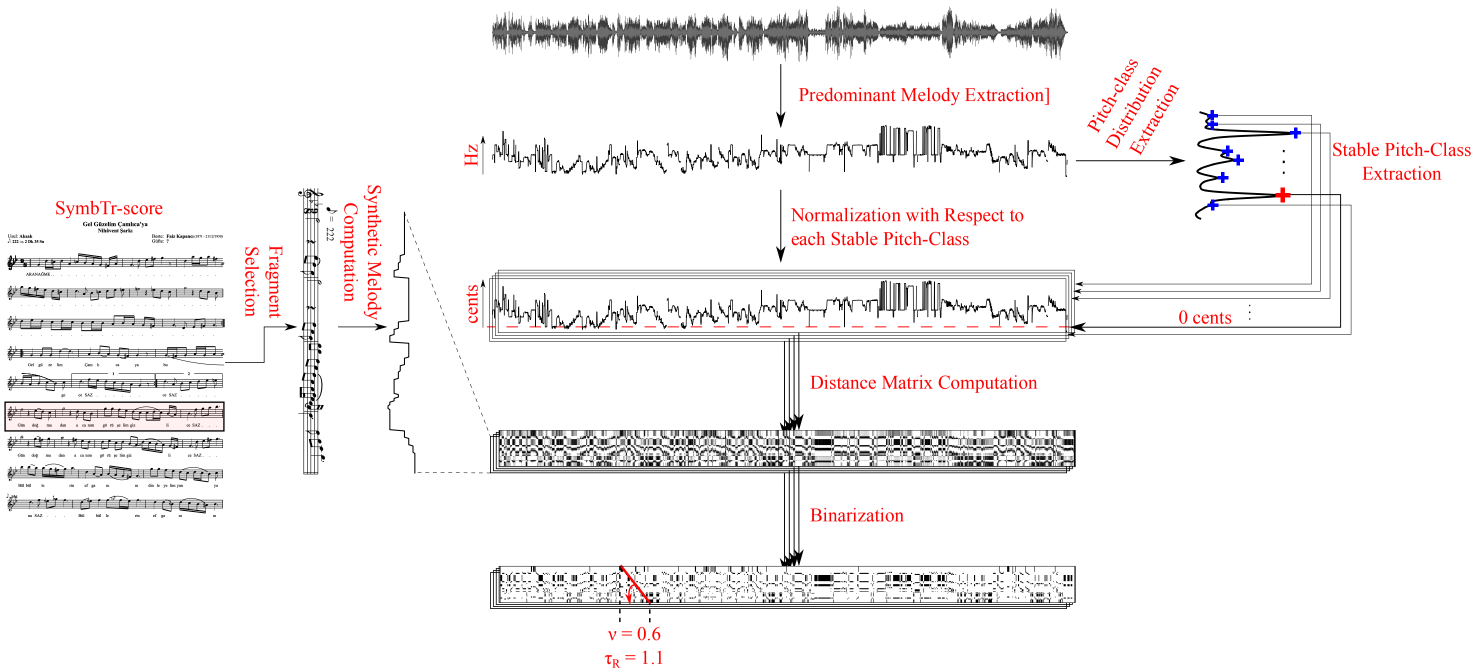

SymbTr-score
Fragment Selection
Synthetic Melody Computation
♪ = 222
Predominant Melody Extraction]
Hz
Pitch-class Distribution Extraction
Stable Pitch-Class Extraction
Normalization with Respect to each Stable Pitch-Class
cents
0 cents
Distance Matrix Computation
Binarization
ν = 0.6
$\tau_R = 1.1$

Figure 6.4: Joint tonic identification and tempo estimation by fragment linking. The weight each stable pitch class is taken as the maximum similarity (Equation 6.10).

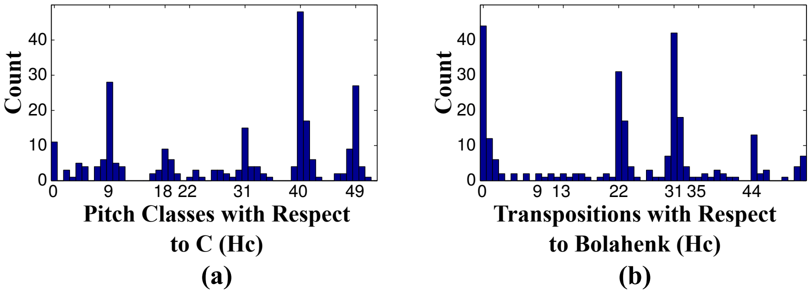


Figure 6.5: Distribution of the annotated tonics in the data collection. a) Pitch class histogram of the annotated tonic with respect to the pitch class C, b) Histogram of the transpositions with respect to *bolahenk*

selceted from the CompMusic audio corpus and **SymbTr** collection, respectively. Some recordings include musical events which do not belong to the composition such as improvisations and even performances of other compositions. The makam of each composition is included in the metadata, which is included in the score (Section 4.1). The pieces cover 28 different makams.[10]

The ground truth is obtained by manually marking the tonic frequency using Makam Toolbox (Gedik & Bozkurt, 2010). Figure 6.5a and Figure 6.5b show the distribution of the annotated tonic with respect to the pitch class C and the distribution of the transpositions with respect to bolahenk, respectively. It can be seen that the annotated tonic are mostly distributed around the semitones with microtonal deviances. Apart from bolahenk, the tonic is mostly performed with a transposition around the perfect fourth, perfect fifth and minor seventh. Nevertheless a considerable number of tonic annotations reside in microtonal pitch classes.

Additional statistics of the **OTMM-section-linking** are given in Section 6.7.3.

6.4.4 Results

Tonic identification by repetitive section linking fails only in one piece (99.2% success rate). In this recording, the vocalist sings

[10] The annotations are available at https://github.com/MTG/otmm_tonic_dataset/tree/2013_ismir.

a gazel (vocal improvisation) in almost three fifth of the duration of the recording with skillful vibratos extending up to ≈ 200 cents peak-to-peak.[11] These vibratos occasionally cross mahur (G5♭) and less frequently reach to gerdaniye (G5), which is in the pitch class of the tonic. Throughout the piece the pitch class G♭ is visited more than G and it shows a wide spread towards G such that no peak is formed in the vicinity of the tonic pitch class. In this case the pitch class G♭ is estimated as the tonic, having a 2.33 Hc deviation.

Using distribution matching, we are able to identify the tonic of 244 performances out of 257 (94.9% success rate). Most of the errors occur in makams Kürdilihicazkar (3 recordings), Muhayyer (3 recordings), Suzidilara (2 recordings), Isfahan and Mahur (1 recordings each), which have complex pitch distributions. The errors are distributed mostly to the fourth (7 recordings) and fifth (4 recordings) of the scale degree. In 4 recordings the tonic is identified as the başlangıç (initial) note, which is the other melodic center of the makam. The average distance between the annotated tonic and the correctly estimated tonics is 0.23 Hc with a standard deviation of 0.21 Hc for both methods.

For comparison, we also modify and test the approach in (Gedik & Bozkurt, 2010) (described in Section 5.7 in detail) using the implementation in Makam Toolbox. We use the `SEN-MEL` as the predominant melody extraction method instead of `YIN`, which was used in the original methodology, in order to improve the pitch estimations. The makam of the piece is provided to the algorithm. We use a subset of the collection with 152 audio recordings. The number of failed identifications is 46, 10 and 1 for audio template distribution matching, score template distribution matching and repetitive section linking, respectively. The results from both of our methods are substantially better than the results obtained from the Makam Toolbox.

6.4.5 Discussion and Summary

We proposed two novel methods that use score information to identify the tonic of audio recording. Assuming the most played pitch

[11] http://dunya.compmusic.upf.edu/makam/recording/f5a89c06-d9bc-4425-a8e6-0f44f7c108ef?start=57387

classes as tonic candidates, the first method compares pitch class distributions computed from the audio and score, and the second method searches for a repetitive score fragments in the audio. We tested the methodologies in a scenario of audio-score collections of OTMM, where the audio and score are already linked with each other at the document level and the score includes the notes, as well as the structural organization, the makam and the tempo of the piece. The results indicate that score information greatly simplifies the tonic identification task. Moreover, the pitch deviationss between the estimated tonic and the annotated tonic are mostly indiscernible. These findings point out the computational potential of knowledge-driven methodologies using multi-modal information. While template distributions computed from audio are similar to the testing distributions with respect to the tuning and limited intonation information in makam level, score distributions indicate these similarities in the (more definitive) composition level. On the other hand, the distribution matching method is still susceptible to the errors seen us audio-based template matching. In the majority of the recordings where distribution matching failed, it was observed that the piece has modulations to pitches that do not belong to the scale of the makam. These contrastive notes and any event can be grouped into characteristic fragments, melodic progressions and structural elements; eventually building the unique the music piece. In general, the lack of such temporal information is the main problem of distribution matching.

By linking repetitive sections, only the tonic of one performance is missed. These results indicate the usefulness of the temporal information in pitch related tasks. The successful results obtained from tonic identification and previously from section linking (Şentürk, Holzapfel, & Serra, 2014) (explained in Section 6.7.4) motivates adapting the "fragment linking" methodology for further computational tasks. In Section 6.6, we work on linking audio and scores in the document level by trying to link sections in each score with corresponding audio recordings. Highly ranked links will indicate the scores and audio recordings related to the same work. We also generalize the tonic identification method to less "complete" scores, where structure information is unknown. The results obtained from the composition identification experiments (Section 6.6.5) show that tonic identification by linking non-repetitive score fragments as short as 8 seconds is possible.

Another interesting direction is to generate predictive models from the scores of each makam. The models can be used to discover characteristic phrases, which could be linked with the audio to further carry relevant tasks such as makam recognition, melodic similarity analysis and expression analysis. Previously we found that multiple viewpoints may be highly predictive in modelling OTMM (Şentürk, 2011). We plan to take advantage of these computational methodologies and models to discover, navigate through and appreciate cultural-specific aspects of makam music of Turkey and other music genres/traditions involving melody-dominant content.

6.5 Score-Informed Tempo Estimation

The relative tempo $\hat{\tau}^{\left(f^{(a)}\right)}$ of an audio fragment $f^{(a)}$ is simply inferred as the ratio between the duration of the aligned audio fragment and the score fragment, and absolute tempo $\tau^{\left(f^{(a)}\right)}$ is computed by multiplying the relative tempo estimation with the nominal tempo $\tau^{\left(f^{(b)}\right)}$ given in the relevant score $f^{(b)}$ as described in Equation 6.4 (Figure 6.2).

In (Holzapfel et al., 2015), the first 20% of the audio recordings is aligned to the first section of the music score using the Hough transform (Section 6.3) to infer the tempo. The annotated tonic $\not{\kappa}^{(a)}$ is used to normalize the audio predominant melody. The tempo of the aligned fragment is used to bias the Bayesian network. This preliminary step is reported to output reliable tempo estimations. The methodology proposed in (Holzapfel et al., 2015) will be explained more in detail in Section 6.7.6.

Later, the average tempo $\tau^{(a)}$ computation is generalized incorporating the process into the score-informed tonic identification method, `SEN-TON`$_{link}$ (Section 6.4). First, the link set, $\Pi^{\kappa^{(a)}}(a, b, f^{(b)}) = \left\{\pi^{\kappa^{(a)}}\left(\bar{f}_1^{(a)}, \bar{f}^{(b)}\right), \pi^{\kappa^{(a)}}\left(\bar{f}_2^{(a)}, \bar{f}^{(b)}\right), \ldots\right\}$, which is computed using the estimated tonic $\kappa^{(a)}$, is selected. The average tempo of the recording is estimated as the tempo $\tau^{\left(\bar{f}_m^{(a)}\right)}$ of the audio fragment $(\bar{f}_m^{(a)})$ in the link set, which emits the highest similarity-value, i.e. $\tau^{(a)} = \underset{\tau\left(\bar{f}_m^{(a)}\right)}{\mathrm{argmax}} \Big\{ \nu^{\kappa^{(a)}}\left(\bar{f}_m^{(a)},\ \bar{f}^{(b)}\right),\ \bar{f}_m^{(a)} \in \bar{\boldsymbol{F}}^{\kappa^{(a)},(a)}(f^{(b)}) \Big\}.$

6.6 Score-Informed Composition Identification

Version identification is an important task in music information retrieval which aims to find the versions of a music piece from a collection of audio recordings automatically (Ellis & Poliner, 2007; Serrà et al., 2009). For popular music such as rap, pop and rock, the task aims to identify the covers of an original audio recording. For classical music traditions a more relevant task is associating compositions with the audio performances. The composition information is highly useful in many other computational tasks such as automatic content description and music discovery (e.g. searching the performances of a composition in a music collection).

For classical music cultures, music collections consisting of music scores and audio recordings along with editorial metadata are desirable in many applications involving cultural heritage archival, music preservation and musicological studies. Composition identification is a crucial step linking performances and compositions during the creation of such music corpus from unlabeled musical data (Thomas et al., 2012).

Composition information can be used to generate and improve linked musical data, enhance the music content description and facilitate navigation in semantic web applications. Consider a scenario, where a musician uploads his interpretation of a composition to a platform such as SoundCloud, YouTube etc. The performed compositions can be automatically identified and labeled semantically using an ontology, e.g. (Raimond, Abdallah, Sandler, & Frederick, 2007). Next the performance can be linked with related concepts (e.g. form, composer, music score) available in other sources such as biographies of the performing artist, the music score of the composition or the musical and editorial metadata stored in open encyclopedias such as MusicBrainz and Wikipedia. Such a scheme would facilitate searching, accessing and navigating relevant music content in a more informed manner. Likewise, tasks such as enhanced listening and music recommendation may also benefit from the musical data linked via automatic composition identification.

Due to inherent characteristics of the oral tradition and the practice of OTMM, performances of the same piece may be substantially different from one another. This aspect brings certain computational challenges for the computational analysis and retrieval of OTMM (Section 2.1). In this Section, a composition identification methodology is presented (Şentürk & Serra, 2016a), which is based on the score-informed tonic identification methodology (Section 6.4). We consider two composition identification scenarios, **1)** identifying the compositions performed in an audio recording, **2)** identifying the audio recordings in which a composition is performed. Note that there might not be any relevant audio recordings for some compositions, and vice versa. The methodology also aims to identify such cases. The contributions can be summarized as:

1. The first composition identification methodology applied to OTMM.
2. An open and editorially complete dataset for composition identification in OTMM (Section 6.6.4).
3. Comparison of Hough transform and `SDTW` in transposition-invariant partial audio-score alignment for OTMM.
4. Simplifications and generalizations of the fragment selection and the fragment duration steps used in the score-informed tonic identification method (Section 6.4) and verification of this method on a larger dataset as a side product of the composition identification experiments (Table 6.1).

For reproducibility purposes, relevant materials such as musical examples, data and results are open and publicly available via the Compmusic Website.[12]

The rest of the Section is structured as follows: Section 6.6.1 gives a definition of the composition identification tasks we are dealing with. Section 6.6.2 explains the methodology applied to both composition identification scenarios explained above. Section 6.6.3 presents the experimental setup, the test dataset and the results. Section 6.6.6 discusses the obtained results and concludes the section with a brief summary.

[12] http://compmusic.upf.edu/node/306

6.6.1 Problem Definition

Given a specific music collection with the set of audio recordings and the set of music scores, two basic composition identification scenarios are:

1. **Composition retrieval:** Identification of the compositions which are performed in an audio recording.
2. **Performance retrieval:** Identification of the audio recordings in which a composition is performed.

These scenarios are ranked retrieval problems where the query is an audio recording and the retrieved documents are the compositions in the composition identification task, and vice versa. In both cases, the common step is to estimate whether a composition and an audio recording are *relevant* to one another. The relevances in the composition identification problems are binary, i.e. 1 if the composition and the audio recordings are paired and 0 otherwise.

The results in both cases can be aggregated by applying this step to multiple documents and queries. Nevertheless, there might be situations where it may be impossible or impractical to retrieve the whole collection, for example restricted access to copyrighted music material or the lack of computational resources in fast-query applications (e.g. real-time composition identification in mobile applications). Moreover, both scenarios might require different constraints to obtain better results and/or process more efficiently. For example, a good performance retrieval method should find multiple relevant audio recordings for a composition; on the other hand only the top ranked documents are important in composition retrieval as more than a single composition is rarely performed in the queried audio recordings (Section 6.6.4). In this paper, we deal with these two tasks separately and leave the joint retrieval task as a future direction to explore.

6.6.2 Methodology

In this method, it is assumed that the scores of the compositions are available. The binary relevance is estimated by partially aligning the score of a composition (b_j) with the audio recording of a performance (a_i). For partial alignment invariant of the transposition of the performance, the score-informed tonic identification procedure described in (Şentürk et al., 2013) (Section 6.4) is used.

The fragment is selected either from the repetitive section or the start of a score. Different fragment durations are tried in the experiments (Section 6.6.3). For partial alignment, either Hough transform or `SDTW` is used with the same parameters given in Section 6.4. For the sake of simplicity, the accumulated weight computation given in Equation 6.9 is replaced by a simpler weight computation by taking the maximum similarity-value for each stable pitch:

$$\nu\left(\phi_k^{(a_i)}\right) = \max\left(\nu^{\phi_k^{(a_i)}}\left(\bar{f}_m^{(a_i)}, \bar{f}^{(b_j)}\right)\right), m \in \left[1 : \left|\Pi^{\phi_k^{(a)}}(a, b, f^{(b)})\right|\right] \tag{6.10}$$

The similarity $\nu(a_i, b_j) \in [0, 1]$ between the audio recording (a_i) and music score (b_j) is taken as the highest weight, i.e. $\nu(a_i, b_j)$ $= \max\left(\nu\left(\phi_k^{(a_i)}\right),\ \phi_k^{(a_i)} \in \mathbf{\Phi}_{PC}^{(a_i)}\right)$. We observe a high similarity value, if the composition (b_j) is indeed performed in the audio recording (a_i) (Section 6.6.2). Note that finding a true pair also implies correctly identifying the tonic pitch class, i.e. $\kappa^{(a_i)} = \operatorname*{argmax}_{\phi_k^{(a_i)}} \nu\left(\phi_k^{(a_i)}\right),\ \phi_k^{(a_i)} \in \mathbf{\Phi}_{PC}^{(a_i)}$. The block diagram of transposition invariant partial-audio score alignment is given in Figure 6.4.

The alignment process is repeated between each audio recording and music score, and a similarity value is obtained for each composition and performance pair in our collection. Figure 6.6 show the similarities computed between the performances in our audio collection (Section 6.6.4) and the composition, "Acemaşiran Peşrevi."[13] In this example, the similarity of the relevant audio recordings are much higher compared to the non-relevant ones. Finally, the performance-composition pairs with low similarity values are discarded using outlier detection (Section 6.6.2), and the relevant pairs are obtained.

[13] http://musicbrainz.org/work/01412a5d-1858-43b3-b5b0-78f383675e9b

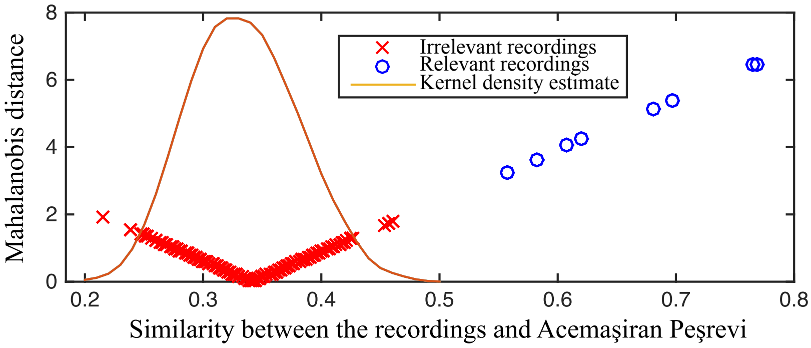


Figure 6.6: Similarity vs Mahalanobis distance between the composition "Acemaşiran Peşrevi" and the audio recordings in the dataset, along with the kernel density-estimate computed from the similarity values between the audio recordings and the composition.

Irrelevant Document Rejection

In many common retrieval scenarios, including composition identification, the users are only interested in checking the top documents (Manning, Raghavan, & Schütze, 2008). After applying partial audio-score alignment between the query and each document, we rank the documents with respect to the similarities obtained. We then reject documents with low similarities according to an automatically learned threshold.

As seen in Figure 6.6, the relevant documents stand as "outliers" among the irrelevant documents with respect to the similarities they emit. To fetch the relevant documents per query, one can apply "outlier detection" using similarities between each document and query. Outlier detection is a common problem, which has many applications such as fraud detection and server malfunction detection (Chandola, Banerjee, & Kumar, 2009).

Upon inspecting the similarity values emitted by irrelevant documents, we have noticed that the values roughly follow a Normal distribution (Figure 6.6). However, the distributions observed for each query have a different mean and variance. This is expected since the similarity computation could be affected by several factors such as the melodic complexities of the score fragment and the audio performance, as well as the quality of the extracted audio predominant melody. To deal with this variability, we compute the Mahalanobis distance (Maesschalck, Jouan-Rimbaud, & Massart, 2000) of each similarity value to the distribution represented by the other similarity values (Figure 6.6).[14] Mahalanobis distance is a unitless and scale-invariant distance metric, which outputs the distance between a point and a distribution in standard deviations.

To reject irrelevant documents we apply a simple method where all documents below a certain threshold are rejected. To learn the decision boundary for thresholding, we apply logistic regression (Manning et al., 2008), a simple binary classification model, to the similarity values and the Mahalanobis distances on labeled data (Section 6.6.4). The training step is explained in Section 6.6.3 in more detail.

After eliminating the documents according to the learned decision boundary, we add a last document called *none* to the end of the list. This document indicates that the query might not have any relevant document in the collection if all of the documents above are irrelevant.

6.6.3 Experiments

In the experiments, two alignment methods (Hough transform vs. SDTW) are compared. We try to align either the repetition in the score as done in (Şentürk et al., 2013) or the start in the score as a simpler alternative and for the case when the structure information is not available in the score. The optimal fragment duration is searched between 4 and 24 seconds.

As mentioned in Section 6.6.1, the performance retrieval and the composition retrieval tasks are evaluated separately. To test the document rejection step, 10-fold cross validation is used. The transposition-invariant partial audio score alignment is applied between each score fragment and audio recording (Section 6.6.2) and then the similarity value for each performance-composition pair in the training set is computed. The Mahalanobis distance is also computed for each query (performance in composition retrieval task and vice versa). Logistic regression is applied to the similarity values

[14] Note that the Mahalanobis distances shown in Figure 6.6 are less than what a "real" Normal distribution would produce. This is because of the contribution by the true pairs to the distribution.

and the Mahalanobis distances computed for each annotated audio-score pair (with the binary relevances 0 or 1), and a decision boundary is learned between the relevant and irrelevant documents. Then, given a query (a composition in the performance retrieval task, and vice versa) from the testing set, we carry out all the steps explained in Section 6.6.2 and reject all the documents (performances in the performance retrieval task, and vice versa) "below" the decision boundary.

Mean average precision (MAP) (Manning et al., 2008) is used to evaluate the methodology. MAP can be considered as a summary of how a method performs for different queries and the number of documents retrieved per query. For the document rejection step, we report the average MAP obtained from the MAPs of each testing set. We also conduct 3-way ANOVA tests on the MAPs obtained from each testing set to find if there are significant differences between the alignment methods, fragment locations and fragment durations. For all results below, the term "significant" has the following meaning: the claim is statistically significant at the $p = 0.01$ level as determined by a multiple comparison test using the Tukey-Kramer statistic.

6.6.4 Dataset

For our experiments, a collection of 743 audio recordings and 146 music scores of different peşrev and sazsemaisi compositions is gathered. The audio recordings are selected from the CompMusic corpus (Uyar et al., 2014). These recordings are either in public-domain or commercially available. The scores are selected from the **SymbTr** score collection (Karaosmanoğlu, 2012). **SymbTr**-scores are given in a machine readable format, which stores the duration and symbol of each note. The structural divisions in the compositions (i.e. the start and end note of each section) and the nominal tempo are also indicated in the scores.

The compositions performed in each audio recording are labeled manually. In the dataset there are 360 recordings associated with 87 music scores, forming 362 audio-score pairs. This information along with other relevant metadata such as the releases, performers and composers are stored in MusicBrainz. Figure 6.7 shows the histogram of the number of relevant compositions per

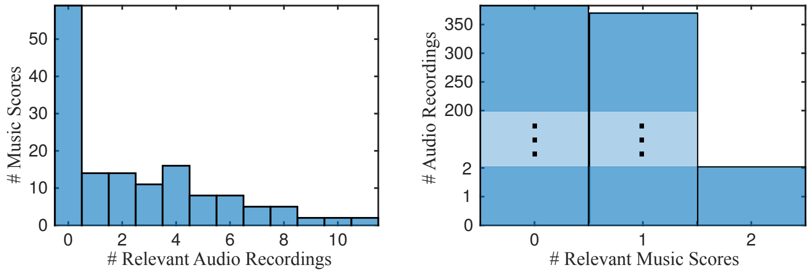


Figure 6.7: The number of relevant documents for the queries a) Histogram of the number of relevant audio recordings per score, b) Histogram of the number of relevant scores per audio recording

audio recording and the number of relevant audio recordings per composition. The number of recordings for a particular composition in our collection may be as many as 11. On the other hand, the releases of OTMM are typically organized such that there is a single composition performed in each track. For this reason, we were only able to obtain two audio recordings in which there are two compositions performed. Note that the tonic frequency changes in the performances of each composition in these two recordings.

The average cardinalities of the compositions per audio recording and audio recordings per composition are 0.49 and 2.48, respectively. Notice that we have also included some compositions in our data collection, which do not have any relevant performances, and vice versa (Figure 6.7). Our methodology also aims to identify such queries without relevant documents. If we consider this case as an additional, special "document" called $none$, the average cardinality for compositions per audio recording and audio recordings per composition is 1.00 and 2.88, respectively.

6.6.5 Results

Before document rejection, the MAP is around 0.47 for both composition retrieval and performance retrieval tasks using either of the alignment methods, fragment locations and fragment durations longer than 8 seconds. The MAP is low before document rejection since the queries without relevant documents will practically have 0 average precision. Figure 6.8 shows the composition retrieval and performance retrieval results before document rejection only for

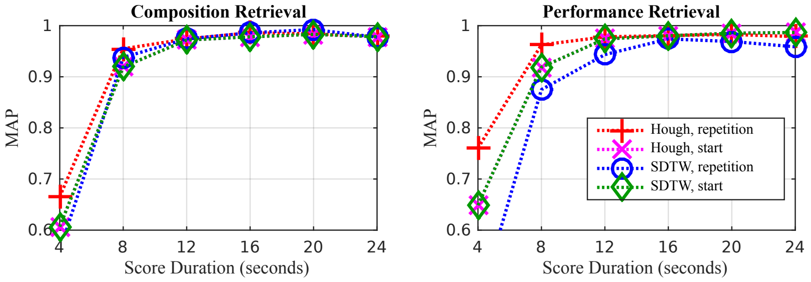


Figure 6.8: MAP for composition and performance retrieval task before document rejection, across different methods, fragment locations and durations. Only the queries with at least one relevant document are considered.

the queries with relevant documents. The retrieval results before document rejection show that most of the audio-score pairs may be found by partial audio-score alignment by using a score fragment of at least 12 seconds. Although Hough transform performs slightly better than SDTW, these increases are not significant for fragment durations longer than 8 seconds.

Figure 6.9 shows the average MAPs from all queries obtained using different fragment durations, fragment locations and partial alignment methods in a 10-fold cross validation scheme. The best average MAP is 0.96 for composition retrieval using either Hough transform or SDTW and aligning 24 seconds from the start. For performance retrieval the best average MAP of 0.95 is achieved using Hough transform and aligning 16 seconds from the start. When we inspect average MAPs obtained from the queries without any relevant documents (Figure 6.10), we observe that the document rejection step always achieves an average MAP higher than 0.95 for all the parameter combinations in the composition retrieval task and an average MAP closer to or higher than 0.9 for all the parameter combinations in the performance retrieval task, respectively.

When we inspect the alignment results, we find that the score fragments were aligned properly for most of the cases. Moreover the tonic is identified almost perfectly for all the audio recordings by aligning the relevant scores (Table 6.1), and we achieved 100% accuracy out of the 362 audio-score pairs by aligning at least 12 seconds from the repetition using Hough transform.

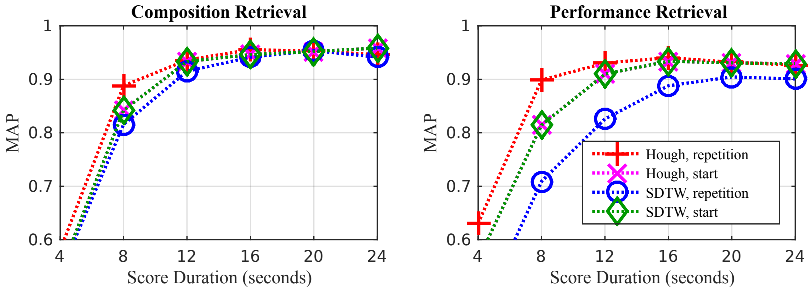


Figure 6.9: MAP for composition and performance retrieval task after document rejection, across different methods, fragment locations and durations. All queries are considered.

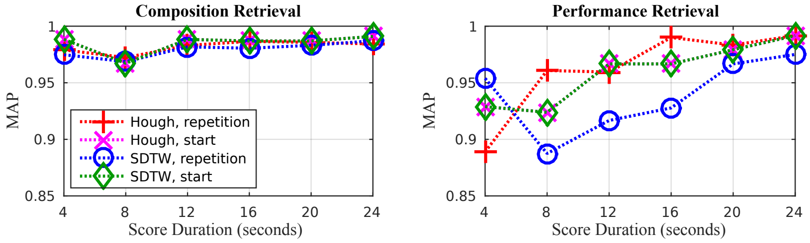


Figure 6.10: MAP for composition and performance retrieval task after document rejection, across different methods, fragment locations and durations. Only the queries with no relevant documents are considered.

Table 6.1: Number of errors in tonic identification

Methods	Locations	**Durations (sec.)**					
		4	8	12	16	20	24
Hough	**Start**	30	15	2	3	2	2
	Repetition	14	5	0	0	0	0
SDTW	**Start**	32	6	3	3	3	3
	Repetition	24	3	1	2	3	3

6.6.6 Discussion and Summary

In this Section, a methodology is proposed to identify the relevant compositions and performances in a collection consisting of audio recordings and music scores, using transposition invariant partial audio-score alignment. The methodology is the first automatic composition identification proposed for OTMM. The methodology is highly successful, achieving 0.95 MAP in retrieving the compositions performed in a recording and 0.96 MAP in retrieving the audio recordings where a composition is performed. What is more, our methodology is not only reliable in identifying relevant compositions and audio recordings but also identifying the cases when there are no relevant documents for a given query. Our algorithm additionally identifies the tonic frequency of the performance of each composition in the audio recording almost perfectly, as a result of partial audio-score alignment.

The results show that even aligning an 8 second fragment is highly effective, nevertheless, the optimal value of fragment duration for composition identification is around 16 seconds. Using a fragment duration longer than 16 seconds is not necessary since it increases the computation time without any significant benefit on identification performance. The results further show that aligning the start is sufficient, and there is no need to exploit the structure information to select a fragment from the repetition as in (Şentürk et al., 2013).

If a fragment of 16 seconds from the start of the score is selected, Hough transform and SDTW produces the same results in both composition retrieval and performance retrieval tasks. One surprising case is the lower MAPs obtained in the performance retrieval task using SDTW to align the repetition. Although the drop is not significant for fragment durations longer than 12 seconds, we observed that SDTW tends to align irrelevant subsequences in the performances with the score fragments, which have similar note-symbol sequences but different durations.

Both Hough transform and SDTW have a complexity of $\left|\boldsymbol{\varrho}^{(a)}\right| \times \left|\hat{\boldsymbol{\Psi}}^{(f^{(b)})}\right|$, where $\left|\boldsymbol{\varrho}^{(a)}\right|$ is the length of the predominant melody extracted from the audio recording (a) and $\left|\hat{\boldsymbol{\Psi}}^{(f^{(b)})}\right|$ is the length of the synthetic melody generated from the score fragment of the composition $(f^{(b)})$. Nonetheless, Hough transform is applied to a sparse, binary similarity matrix, hence it can operate faster than SDTW. Moreover, Hough transform is a simpler algorithm. These properties make Hough transform a cheaper and effective alternative to more complex alignment algorithms such as SDTW, when precision in intra-alignment (e.g. note-level) is not necessary. Given these observations, we select alignment of the first 16 seconds of the score using Hough transform as the optimal setting. As the next step we would like to evaluate the method on more forms, possibly with shorter structural elements such as the vocal form, şarkı. We would also like to investigate network analysis methods to identify the relevant performances and compositions jointly.

For the score fragments longer than 8 seconds, the tonic identification errors always occur in two historical recordings, where the recording speed (hence the pitch) is not stable and another recording where the musicians sometimes play the repetition by transposing the melodic intervals by a fifth. Even though the tonic identification has failed in these cases, the fragments are correctly aligned to the score. For such recordings, the stability of the tonic frequency can be assessed and the tonic frequency can be refined locally by referring to aligned tonic notes in the alignment path computed using SDTW.

From Figure 6.9, we can observe that by using a simple outlier detection step based on logistic regression, we were able to reject most of the irrelevant documents in both composition retrieval and performance retrieval scenarios. By comparing Figure 6.8 with Figure 6.9, we can also conclude that this step does not remove many relevant documents, providing reliable performance and composition matches. The usefulness of this step is more evident when the results for the queries with no relevant documents are checked (Figure 6.10). For such queries, since all the documents typically have a low, comparable similarity, our methodology is able to reject almost all the irrelevant documents. From Figure 6.10, we can also observe that the document rejection step is robust to changes in the fragment duration, the fragment location and the alignment method.

The method can easily be adapted to neighboring music cultures such as Greek, Armenian, Azerbaijani, Arabic and Persian music, which share similar melodic characteristics. We hope that our method would be a starting point for future studies in automatic composition identification, and facilitate future research and applications on linked data, automatic music description, discovery and archival.

6.7 Section Linking

In this Section, we focus on marking the time intervals in the audio recording of a piece with the musically relevant structural elements (sections) marked in the score of the same piece (or briefly "section linking"). The proposed method extracts features from the audio recording and the sections in the score. From these features, similarity matrices are computed for each section. The method applies Hough transform (Duda & Hart, 1972) to the similarity matrices in order to detect section candidates. Then, it selects between these candidates by searching through the paths, which reflect the sequence of sections implied by the musical form, in a directed acyclic graph (DAG) directed acyclic graph. We optimize the method for the cultural-specific aspects of OTMM. By *linking* score sections with the corresponding fragments in the audio recordings, computational operations that are specific to this type of music, such as *makam* recognition (Gedik & Bozkurt, 2010), tuning analysis (Bozkurt et al., 2009) and rhythm analysis can be done at the section level, providing a deeper insight into the structural, melodic or metrical properties of the music.

Section linking has been studied in two papers (Şentürk et al., 2012; Şentürk, Holzapfel, & Serra, 2014) in the scope of the thesis. This Section focuses on the section linking methodology explained in (Şentürk, Holzapfel, & Serra, 2014). The preliminary methodology (Şentürk et al., 2012) is explained in Appendix A separately for the sake of brevity. Until the end of Section 6.7, an audio recording[15] of the composition *Şedaraban Sazsemaisi*[16] is used for illustration.

The remainder of the Section is structured as follows: Section 6.7.1 makes a formal definition of *section linking*. Section 6.7.2 explains the proposed methodology in detail. Section 6.7.4 presents the experiments carried out to evaluate the method. Section 6.7.3 describes the dataset used to test the methodology. Section 6.7.5 presents the results obtained from the experiments and Section 6.7.7 provides a discussion and a brief conclusion.

[15] http://musicbrainz.org/recording/efae832f-1b2c-4e3f-b7e6-62e08353b9b4

[16] http://musicbrainz.org/work/1eb2ca1e-249b-424c-9ff5-0e1561590257

6.7.1 Problem Definition

We define *section linking* as “marking the time intervals in the audio recording at which musically relevant structural elements (sections) given in the score are performed.” In this task, we start with a score and an audio recording of a music piece. The score and audio recording are known to be related with the same work (composition) via available metadata, i.e. they are already linked with each other in the document-level.

The score includes the notes, and it is divided into sections, some of which are repeated. These sections are annotated; and the label, the start and end of each section are provided in the score, including the compositional repetitions. Therefore, we do not need any structural analysis to find the structural elements. Later in Section 6.12, the semiotic section labels obtained in score structure analysis (Section 4.3) are used instead. From the start and end of each section, the sequence of the sections are known. The tempo and the makam of the piece are also available in the score. The audio recording follows the section sequence given in the score with possible section insertions, omissions, repetitions and substitutions. The tonic of the audio recording is known, etiher by manually annotating or by automatic tonic identification (Section 6.4). For the sake of brevity, the tonic symbol is omitted (e.g. the similarity matrix $\boldsymbol{B}^{\kappa^{(a)},(a,\bar{s}_j^{(b)})}$ is written as $\boldsymbol{B}^{(a,\bar{s}_j^{(b)})}$). The performance might include various expressive decisions such as musical material that are not related to the piece, phrase repetitions/omissions, pitch deviations.

Following the definitions introduced in Section 4.3.1, we apply our method to obtain the (estimated) audio section sequence $\bar{\mathbf{S}}^{(a)}$ in the audio recording, where each section, $\bar{s}_i^{(a)} = \left\langle \bar{\mathbf{N}}^{\left(\bar{s}_i^{(a)}\right)}, s_i^{(a)},\ t(\bar{s}_i^{(a)})\right\rangle$, in the sequence is paired with a section label $s_i^{(a)} \in \mathcal{S}^{(b)} = \left\{\mathcal{S}_s{}^{(b)}, unrelated\right\}$ in the composition. Ideally, the *true audio section sequence*, $\bar{\mathfrak{S}}^{(a)}$, and *section link sequence*, $\bar{\mathbf{S}}^{(a)}$ should be identical.

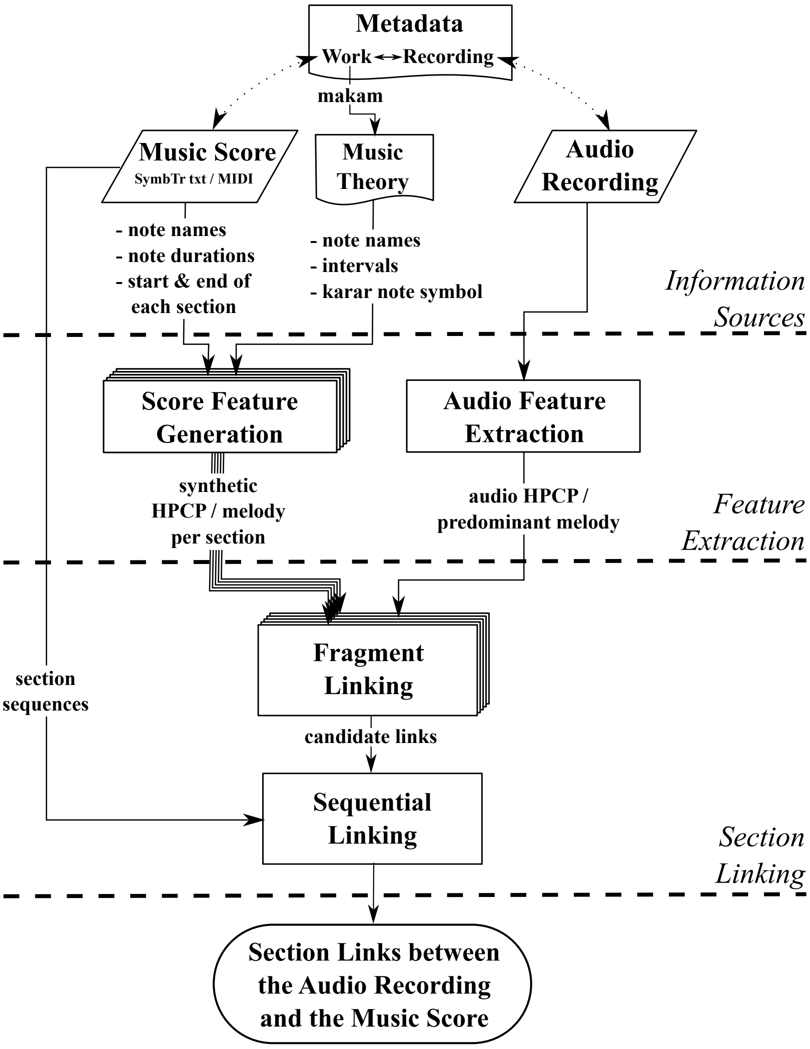


Figure 6.11: Block Diagram of the Section Linking Methodology.

6.7.2 Methodology

By incorporating makam music knowledge, and considering culture-specific aspects of the makam music practice (such as pitch deviations and heterophony), we specialize the section linking methodology to OTMM. Given the score representation (b) of a composition and the audio recording (a) of the performance of the same composition, the procedure to link the sections of a score with the corresponding sections in the audio recording is as follows (Figure 6.11):

1. From music-theory knowledge, a dictionary is generated consisting $\langle makam,\ karar \rangle$ pairs, which stores the karar of each makam (e.g. if the makam of the piece is Hicaz, the karar is A4.). The karar note is used as the reference symbol during the generation of score features for each section (Section 6.2). We also apply the theoretical intervals for a makam as defined in AEU theory to generate the score features from the machine-readable score (Section 6.2).

2. Features are computed from the audio recording (a) and the musically relevant sections $(\bar{s}_j^{(b)})$ with unique section labels $(\forall s_j^{(b)} \in \mathcal{S}_s{}^{(b)})$ of the score (b) (Section 6.2). If **SymbTr**-MIDI scores are used, audio HPCPs $\hat{\boldsymbol{\Gamma}}^{(a)}$ and synthetic HPCPs $\hat{\boldsymbol{\Omega}}^{(b)}$ are computed for the audio recording and the music score, respectively. If **SymbTr**-txt scores are used, predominant melody $\boldsymbol{\varrho}^{(a)}$ and synthetic melody $\hat{\boldsymbol{\Psi}}^{(b)}$ are computed for the audio recording and the music score, respectively.

3. A similarity matrix $\boldsymbol{B}^{(a,\bar{s}_j^{(b)})}$ is computed for each section $(\bar{s}_j^{(b)})$, measuring the similarity between the score features of the particular section and the audio features of the whole recording (Section 6.3). By applying Hough transform to the similarity matrices, candidate links $\pi(\bar{s}_i^{(a)}, \bar{s}_j^{(b)})$, where $s_i^{(a)} = s_j^{(b)} \in \mathcal{S}_s{}^{(b)}$, are estimated in the audio recording for each section given in the score (Section 6.3.1).

   In order to restrict the angles searched in Hough transform to an interval $[\theta_{min},\ \theta_{max}]$, the relative tempo of all the true sections $\hat{\tau}^{\left(\bar{\mathfrak{s}}_i^{(a)}\right)}$ in the section linking dataset (see Section 6.7.3) are computed. The relative tempo $\hat{\tau}^{\left(\bar{s}_i^{(a)}\right)}$ of a section candidate is restricted between 0.5 and 1.5, covering most of the observed tempo distribution. This limits the searched angles in Hough transform between $\theta_{min} = 27°$ and $\theta_{max} = 56°$ (Equation 6.5).

   Figure 6.2 shows the candidate link estimation process for the Teslim section of *Şedaraban Sazsemaisi*.

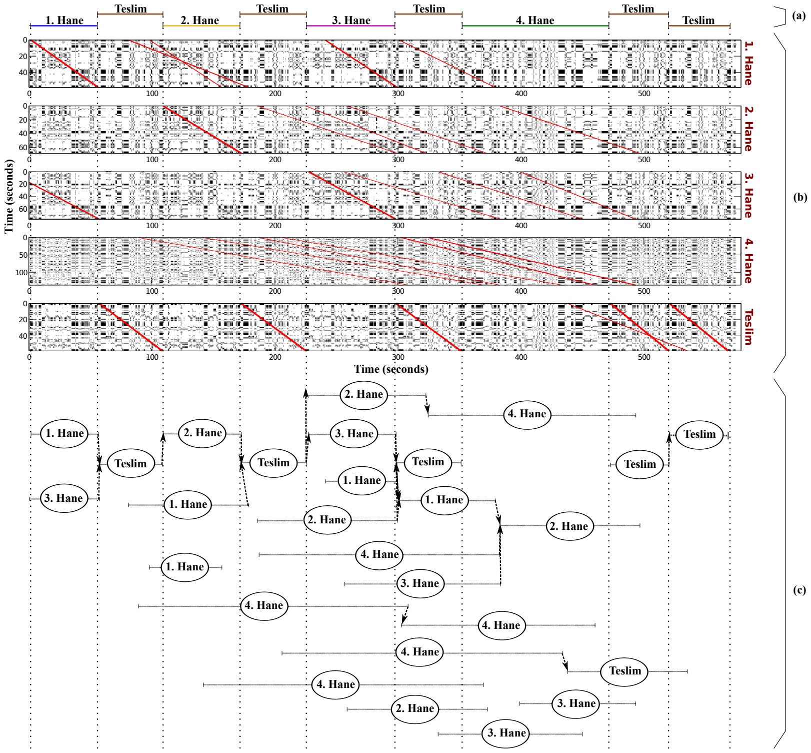


Figure 6.12: Extraction of all possible paths from the estimated candidates in an audio recording of *Şedaraban Sazsemaisi*. a) Annotated Sections, b) Candidate Estimation, c) The directed acyclic graph formed from the candidate links.

4. Treating the candidate links as labeled nodes, a directed acyclic graph (DAG) is generated. Using section sequence information ($\bar{\mathbf{S}}^{(b)}$) given in the score, all possible paths in the DAG are searched and the most-likely candidates are identified. Then, the non-estimated time intervals are guessed. The final links are marked as section links Figure 6.12. We now proceed to explained this step (termed as *sequential linking*) in detail.

Sequential Linking

By inspecting Figures 6.12a and 6.12b, it can be seen that all ground truth annotations are among the detected candidates, with problems in the alignment of 4th Hane. However, as there are also many false positives, we use knowledge about the structure of the composition to improve the candidate selection. Considering the candidate links as nodes in a DAG, we first extract all possible paths from the DAG according to the score section symbol sequence $\bar{\mathbf{S}}^{(b)} = \left[\bar{s}_1^{(b)}, \ldots, \bar{s}_{|\bar{\mathbf{S}}^{(b)}|}\right]$. We then decide the most likely paths. Finally, we attempt to guess non-estimated time intervals in the audio and obtain the final section links.

Path Extraction: Having obtained the candidate section links $\pi(\bar{s}_i^{(a)}, \bar{s}_j^{(b)})$, each section candidate $\bar{s}_i^{(a)}$, may be interpreted as a node. A node has the following labels:

- Section symbol, $s_i^{(a)} = s_j^{(b)} \in \mathcal{S}_s{}^{(b)}$
- Time interval $t(\bar{s}_i^{(a)}) = \left[t_{ini}(\bar{s}_i^{(a)})\ t_{fin}(\bar{s}_i^{(a)})\right]$.
- Relative tempo, $\hat{\tau}^{(\bar{s}_i^{(a)})}$, with its value restricted according to the duration constraint, i.e. to the interval $[0.5,\ 1.5]$.
- Weight $\nu(\bar{s}_i^{(a)})$, equal to the similarity $\nu(\bar{s}_i^{(a)}, \bar{s}_j^{(b)})$ of the candidate section link $\pi(\bar{s}_i^{(a)}, \bar{s}_j^{(b)})$, in the interval $[0, 1]$ (see Section 6.3.3).

If the *final time* of a node, $t_{fin}(\bar{s}_k^{(a)})$, is close enough to the *initial time* of another node, $t_{ini}(\bar{s}_i^{(a)})$, i.e. $|t_{fin}(\bar{s}_k^{(a)}) - t_{ini}(\bar{s}_i^{(a)})| < \alpha$ (α is chosen as 3 seconds), a directed edge $e_{k \to i} = \left\langle \bar{s}_k^{(a)}, \bar{s}_i^{(a)} \right\rangle$ from $\bar{s}_k^{(a)}$ to $\bar{s}_i^{(a)}$ is formed. The nodes and edges form a DAG, $\boldsymbol{\mathcal{G}}$ (Figure 6.12c).

We define a path $\mathbf{p}_i$ as a sequence of nodes $\bar{\mathbf{S}}^{(\mathbf{p}_i)} = \left[\bar{s}_1^{(\mathbf{p}_i)}, \ldots, \bar{s}_{|\bar{\mathbf{S}}^{(\mathbf{p}_i)}|}^{(\mathbf{p}_i)}\right] \subset \boldsymbol{\mathcal{N}}(\boldsymbol{\mathcal{G}})$, where $\boldsymbol{\mathcal{N}}(\boldsymbol{\mathcal{G}})$ denotes the node set of the graph; and weighted edges $\mathbf{E}^{(\mathbf{p}_i)} = \left[e_1^{(\mathbf{p}_i)}, e_2^{(\mathbf{p}_i)}, \ldots, e_k^{(\mathbf{p}_i)}, \ldots, e_{K_i-1}^{(\mathbf{p}_i)}\right] \subset \boldsymbol{\mathcal{E}}(\boldsymbol{\mathcal{G}})$, where $e_k^{(\mathbf{p}_i)}$ represents the directed edge $e_{k\to k+1}^{(\mathbf{p}_i)} = \left\langle \bar{s}_k^{(\mathbf{p}_i)}, \bar{s}_{k+1}^{(\mathbf{p}_i)} \right\rangle$ and $\boldsymbol{\mathcal{E}}(\boldsymbol{\mathcal{G}})$ denotes the edge set of the graph. The length of the path is $|\mathbf{p}_i| = \left|\mathbf{E}^{(\mathbf{p}_i)}\right| = |\bar{\mathbf{S}}^{(\mathbf{p}_i)}| - 1$. We also obtain the section symbol sequence $\mathbf{S}^{(\mathbf{p}_i)} = \left[s_1^{(\mathbf{p}_i)}, \ldots, s_{|\mathbf{S}^{(\mathbf{p}_i)}|}^{(\mathbf{p}_i)}\right]$, where $k \in [1 : |\mathbf{S}^{(\mathbf{p}_i)}|]$ and $s_k^{(\mathbf{p}_i)} \in \mathcal{S}_s$ is the section label of the node, $\bar{s}_k^{(\mathbf{p}_i)}$.

To track the section sequences in audio with reference to the score section symbol sequence $\mathbf{S}^{(b)}$, we construct a variable-length Markov model (VLMM) (Bühlmann & Wyner, 1999). A VLMM is an ensemble of Markov models from an order of 1 to a maximum order of N_{max}. Given a section symbol sequence $\mathbf{S}^{(\mathbf{p}_i)}$, the transition probability $\xi_{k-1}^{(\mathbf{p}_i)}$ of the edge $e_{k-1}^{(\mathbf{p}_i)}$ is computed as:

$$\xi_{k-1}^{(\mathbf{p}_i)} = \Pr\left(s_k^{(\mathbf{p}_i)},\ s_{k-1}^{(\mathbf{p}_i)} \dots s_{k-n}^{(\mathbf{p}_i)}\right), \quad n = \min\left(N_{max}, k-1\right) \tag{6.11}$$

where $\Pr\left(x_k,\ x_{k-1} \dots x_{k-n}\right)$ is the conditional probability of the event x_k occuring after the sequence of events $x_{k-1} \dots x_{k-n}$ (x_{k-n} occurs the first).

In the dataset, the sections are repeated at most twice in succession (Section 6.7.3). Hence, the maximum order of the model N_{max} is chosen as 3, which is necessary and sufficient to track the position of the section sequence. VLMMs are trained from the score section symbol sequences, $s^{(b)}$, and true audio section symbol sequences, $\bar{\mathfrak{S}}$, of other audio recordings whose compositions are built from a common symbol set $\mathcal{S}_s{}^{(b)}$. If a composition is performed partially in an audio recording, the recording is not used for training.

If a node $\bar{s}_k^{(a)}$ has outgoing but no incoming edges, it is the starting node of a path. A node $\bar{s}_k^{(a)}$ is connectable to the end of a path $\mathbf{p}_i$, if the following conditions are satisfied:

i. A directed edge $e_{|\mathbf{p}_i|+1\to k}^{(\mathbf{p}_i)}$ from $\bar{s}_{|\bar{\mathbf{S}}^{(\mathbf{p}_i)}|}^{(\mathbf{p}_i)}$ to $\bar{s}_k^{(a)}$ exists, i.e.

$$\left| t_{fin}(\bar{s}_{|\bar{\mathbf{S}}^{(\mathbf{p}_i)}|}^{(\mathbf{p}_i)}) - t_{ini}(\bar{s}_k^{(a)}) \right| < \alpha, \quad \alpha = 3 \text{ seconds.}$$

ii. The transition probability from $\bar{s}_{|\bar{\mathbf{S}}^{(\mathbf{p}_i)}|}^{(\mathbf{p}_i)}$ to $\bar{s}_k^{(a)}$ is greater than zero, i.e. $\Pr\left(s_k^{(a)},\ s_{|\bar{\mathbf{S}}^{(\mathbf{p}_i)}|}^{(\mathbf{p}_i)} \dots s_{|\bar{\mathbf{S}}^{(\mathbf{p}_i)}|-n+1}^{(\mathbf{p}_i)}\right) > 0, \quad n = \min\left(N_{max},\right.$ $\left.|\bar{\mathbf{S}}^{(\mathbf{p}_i)}|\right)$.

Starting from the nodes with no incoming edges, we iteratively build all paths in the graph by applying the above rules. While traversing the nodes, an additional path is encountered, if:

- A node in the path is connectable to more than one node. There exists a path for each of these connectable nodes. All these paths share the same starting node.

- The transition probability of an edge to the node $\bar{s}_k^{(a)}$ is zero for the current path $\mathbf{p}_i$, i.e. $\left| t_{fin}(\bar{s}^{(\mathbf{p}_i)}_{|\bar{\mathbf{S}}^{(\mathbf{p}_i)}|}) - t_{ini}(\bar{s}_k^{(a)}) \right| < \alpha, \quad \alpha =$ 3 seconds, and $\Pr\left(s_k^{(a)},\ s^{(\mathbf{p}_i)}_{|\bar{\mathbf{S}}^{(\mathbf{p}_i)}|} \dots s^{(\mathbf{p}_i)}_{|\bar{\mathbf{S}}^{(\mathbf{p}_i)}|-n+1} \right) = 0, \quad n = \min\left(N_{max}, |\bar{\mathbf{S}}^{(\mathbf{p}_i)}|\right)$, but the transition probability is greater than zero for a VLMM with a smaller order $0 < n' < n$. In this case, there exists a path that has $\bar{s}^{(\mathbf{p}_i)}_{|\bar{\mathbf{S}}^{(\mathbf{p}_i)}|-n'+1}$ as the starting node.

Traversing the nodes and edges, we obtain all possible paths $\mathcal{P}(\mathcal{G}) = \left\{ \mathbf{p}_1, \dots, \mathbf{p}_{|\mathcal{P}(\mathcal{G})|} \right\}$ from the candidate links, where $|\mathcal{P}(\mathcal{G})|$ is the total number of paths (Figure 6.13a). The total weight of a path $\mathbf{p}_i$ is calculated by adding the weights of the nodes and the transition probabilities of the edges forming the path:

$$\nu(\mathbf{p}_i) = \sum_{k=1}^{|\mathbf{S}^{(\mathbf{p}_i)}|} \nu(\bar{s}_k^{(\mathbf{p}_i)}) + \sum_{k=1}^{|\mathbf{p}_i|} \xi_k^{(\mathbf{p}_i)} \tag{6.12}$$

In summary, each path $\mathbf{p}_i$ has the following labels:

- A sequence of labeled nodes, $\bar{\mathbf{S}}^{(\mathbf{p}_i)} \subset \mathcal{N}(\mathcal{G})$, $\left|\bar{\mathbf{S}}^{(\mathbf{p}_i)}\right| = |\mathbf{p}_i| + 1$, representing the sections.

- Directed, labeled edges connecting the nodes, $\mathbf{E}^{(\mathbf{p}_i)} \subset \mathcal{E}(\mathcal{G})$, $\left|\mathbf{E}^{(\mathbf{p}_i)}\right| = |\mathbf{S}^{(\mathbf{p}_i)}|$.

- Section symbol sequence, $\mathbf{S}^{(\mathbf{p}_i)} = [s_1^{(\mathbf{p}_i)}, \dots, s^{(\mathbf{p}_i)}_{|\bar{\mathbf{S}}^{(\mathbf{p}_i)}|}]$.

- Time interval $t(\mathbf{p}_i) = [t_{ini}(\mathbf{p}_i)\ \ t_{fin}(\mathbf{p}_i)]$, where $t_{ini}(\mathbf{p}_i) = t_{ini}\left(\bar{s}_1^{(\mathbf{p}_i)}\right)$ denotes the initial time and $t_{fin}(\mathbf{p}_i) = t_{fin}\left(\bar{s}^{(\mathbf{p}_i)}_{|\bar{\mathbf{S}}^{(\mathbf{p}_i)}|}\right)$ denotes the final time of the path.

- Total weight, $\nu(\mathbf{p}_i)$.

Elimination of Improbable Candidates: Correct paths usually have a greater number of nodes (and edges) as depicted in

Figure 6.13a. Moreover, the correct nodes typically have a higher weight than the others. Therefore, the correct paths have a higher total weight than other paths within their duration. Assuming $\mathbf{p}^*$ is the path with the highest total weight, we remove all other nodes within the duration of the path $[t_{ini}(\mathbf{p}^*)\ t_{fin}(\mathbf{p}^*)]$ (Algorithm 3, Figure 6.13b,d). Notice that $\mathbf{p}^*$ can remove one or more nodes from the "middle" of another path, which has a longer time duration than $\mathbf{p}^*$; effectively removing edges, splitting the path into two, and hence creating two separate paths.

Algorithm 3 Remove overlapping nodes

```
function remove_overlap($\mathcal{N}(\mathcal{G})$, $\mathbf{p}^*$)
    $\mathcal{N}_{\text{chk}} \Leftarrow \mathcal{N}(\mathcal{G}) - \bar{\mathbf{S}}^{(\mathbf{p}^*)}$
    for $\bar{s}_k \in \mathcal{N}_{\text{chk}}$ do
        if $[t_{ini}(\mathbf{p}^*)\ t_{fin}(\mathbf{p}^*)] \cap [t_{ini}(\bar{s}_k)\ t_{fin}(\bar{s}_k)] > 3$ seconds
then
            $\mathcal{N}(\mathcal{G}) \Leftarrow \mathcal{N}(\mathcal{G}) - \bar{s}_k$
    return $\mathcal{N}(\mathcal{G})$
```

After removing the nodes within the time interval covered by the path $\mathbf{p}^*$, the related section sequence $\bar{\mathbf{S}}^{(\mathbf{p}^*)}$ becomes unique within this time interval, and are therefore considered final section links. The section symbol sequence of the path $\mathbf{S}^{(\mathbf{p}^*)}$ follows a *score section symbol subsequence* $\mathbf{S}^{(b^*)} = \left[s_j^{(b)}, \ldots, s_k^{(b)}\right]$ of the score section symbol sequence $\mathbf{S}^{(b)} = \left[s_1^{(b)}, \ldots, s_j^{(b)}, \ldots, s_k^{(b)}, \ldots, s_{|\mathbf{S}^{(b)}|}^{(b)}\right]$, $1 \leq j \leq k \leq |\mathbf{S}^{(b)}|$. Next, we remove inconsequent nodes occurring before and after the audio section sequence, $\mathbf{p}_i$ with respect to $\mathbf{S}^{(b^*)}$ (see Algorithm 4).

We define two score section symbol subsequences $\mathbf{S}^{(b^-)}$ and $\mathbf{S}^{(b^+)}$, which occur before and after $\mathbf{S}^{(b^*)}$, respectively. Since the sections may be repeated twice in succession within a performance (Section 6.7.3), they depend on the first two section symbols, $\{s_1^{(\mathbf{p}^*)}, s_2^{(\mathbf{p}^*)}\}$, and the last two section symbols, $\{s_{|\mathbf{S}^{(b^*)}|-1}^{(\mathbf{p}^*)}, s_{|\mathbf{S}^{(b^*)}|}^{(\mathbf{p}^*)}\}$, of the section symbol sequence $\mathbf{S}^{(\mathbf{p}^*)}$ of the path $\mathbf{p}^*$:

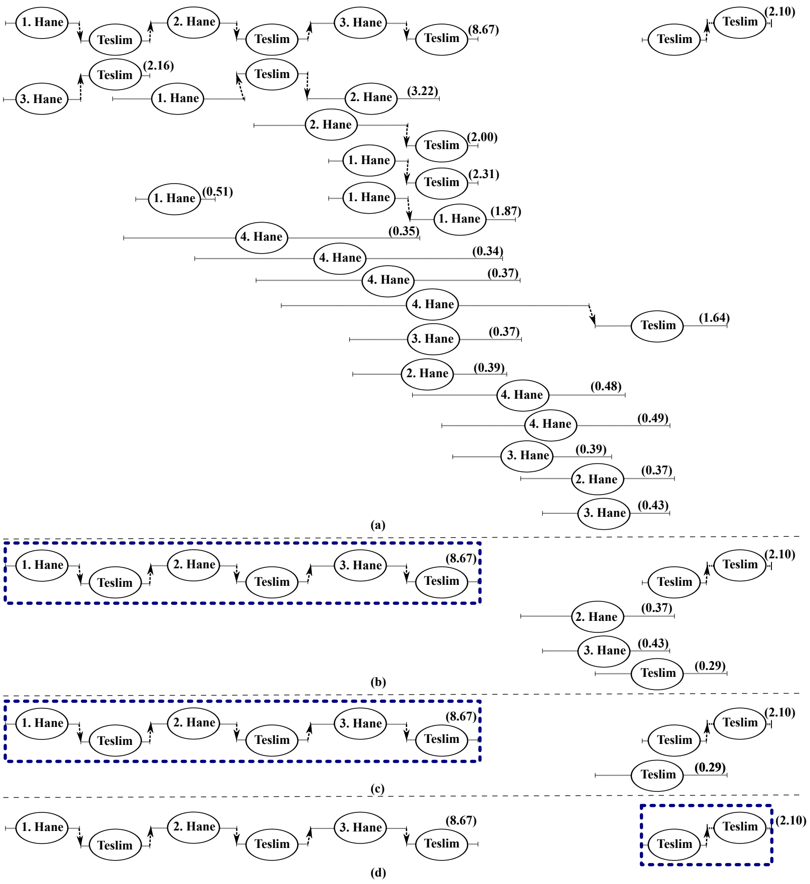


Figure 6.13: Graphical example for the sequential linking for the *Şedaraban Sazsemaisi*. a) All possible paths extracted from the graph. The number in parenthesis in the right side of each path indicates the total weight of the path. b) Overlapping nodes with respect to the path with the highest weight are removed (see Algorithm 3). c) Inconsequent nodes with respect to the path with the highest weight are removed (see Algorithm 4). d) Overlapping node with respect to the path with the second highest weight is removed.

$$\mathbf{S}^{(b^-)} = \begin{cases} \emptyset, & s_1^{(\mathbf{p}^*)} = s_2^{(\mathbf{p}^*)} = s_1^{(b)} \\ \left[s_1^{(b)}, \ldots, s_{j-1}^{(b)}\right], & s_1^{(\mathbf{p}^*)} = s_2^{(\mathbf{p}^*)} \neq s_1^{(b)} \\ \left[s_1^{(b)}, \ldots, s_j^{(b)}\right], & s_1^{(\mathbf{p}^*)} \neq s_2^{(\mathbf{p}^*)} \end{cases} \tag{6.13}$$

$$\mathbf{S}^{(b^+)} = \begin{cases} \emptyset, & s^{(\mathbf{p}^*)}_{|\mathbf{S}^{(b^*)}|-1} = s^{(\mathbf{p}^*)}_{|\mathbf{S}^{(b^*)}|} = s^{(b)}_M \\ \left[s^{(b)}_{k+1}, \dots, s^{(b)}_M\right], & s^{(\mathbf{p}^*)}_{|\mathbf{S}^{(b^*)}|-1} = s^{(\mathbf{p}^*)}_{|\mathbf{S}^{(b^*)}|} \neq s^{(b)}_M \\ \left[s^{(b)}_{k}, \dots, s^{(b)}_M\right], & s^{(\mathbf{p}^*)}_{|\mathbf{S}^{(b^*)}|-1} \neq s^{(\mathbf{p}^*)}_{|\mathbf{S}^{(b^*)}|} \end{cases} \tag{6.14}$$

Since sections given in the $\mathbf{S}^{(b^-)}$ and $\mathbf{S}^{(b^+)}$ have to be played in the audio before and after $\mathbf{S}^{(\mathbf{p}^*)}$ respectively, we may remove all the nodes occurring before and after $\mathbf{p}^*$, which do not follow these score section symbol subsequences (Algorithm 4, Figure 6.13c).

Algorithm 4 Remove inconsequent nodes

```
function remove_inconsequent($\mathcal{N}(\mathcal{G})$, $\mathbf{p}^*$)
    $\mathcal{N}_{\text{chk}} \Leftarrow \mathcal{N}(\mathcal{G}) - \bar{\mathbf{S}}^{(\mathbf{p}^*)}$
    $\mathbf{S}^{(b^-)} \Leftarrow$ get_prev_sec_subseq($\mathbf{S}^{(\mathbf{p}^*)}, \mathbf{S}^{(b^*)}$)   ▷ Equation 6.13
    $\mathbf{S}^{(b^+)} \Leftarrow$ get_next_sec_subseq($\mathbf{S}^{(\mathbf{p}^*)}, \mathbf{S}^{(b^*)}$)   ▷ Equation 6.14
    for $\bar{s}_k \in \mathcal{N}_{\text{chk}}$ do
        if $t_{ini}(\bar{s}_k) < t_{ini}(\mathbf{p}^*)$   &   $s_k \notin \mathbf{S}^{(b^-)}$ then
            $\mathcal{N}(\mathcal{G}) \Leftarrow \mathcal{N}(\mathcal{G}) - \bar{s}_k$
        else if $t_{fin}(\bar{s}_k) > t_{fin}(\mathbf{p}^*)$   &   $s_k \notin \mathbf{S}^{(b^+)}$ then
            $\mathcal{N}(\mathcal{G}) \Leftarrow \mathcal{N}(\mathcal{G}) - \bar{s}_k$
    return $\mathcal{N}(\mathcal{G})$
```

In order to obtain the optimal (estimated) audio section sequence $\bar{\mathbf{S}}^{(a)}$, we iterate through the paths ordered by weight $\nu(\mathbf{p}_i)$ and remove improbable nodes according to this path by using Algorithms 3 and 4. Note that the final sequence might be fragmented into several disconnected paths, as shown *e.g.* in Figure 6.13d. The final step of our algorithm attempts to fill these gaps based solely on information about the compositional structure.

Guessing non-linked time intervals: After we obtained a list of links based on audio and structural information, there might be some time intervals where there are no sections linked (Figure 6.13d).

Assume that the time interval $t^* = [t^*_{ini}\ t^*_{fin}]$ is not linked and it lies between two paths, $\{\mathbf{p}^-, \mathbf{p}^+\}$, before and after the non-linked interval. Note that the path $\mathbf{p}^-$ or $\mathbf{p}^+$ can be empty, if the time interval is in the start or the end of the audio recordings, respectively.

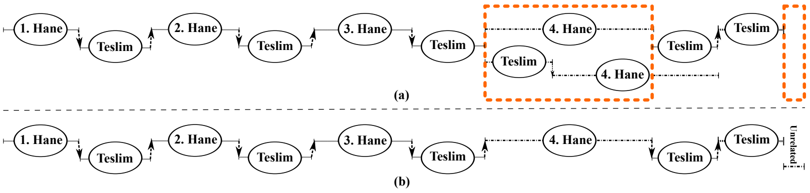


Figure 6.14: Guessing non-estimated time intervals shown on an audio recording of *Şedaraban Sazsemaisi* a) Possible paths computed with respect to the median of the relative tempos of all nodes. b) Final links.

These paths would follow the score section symbol subsequences $\mathbf{S}^{(b^-)}$ and $\mathbf{S}^{(b^+)}$, respectively, and there will be a score section symbol subsequence $\mathbf{S}^{(b^*)} = [s_1^{(b^*)}, \ldots, s_{|\mathbf{S}^{(b^*)}|}^{(b^*)}]$, lying between $\mathbf{S}^{(b^-)}$ and $\mathbf{S}^{(b^+)}$. This score symbol subsequence can be covered in the time interval t^*. Since the sections may be repeated twice in succession within a performance (Section 6.7.3), the first and the last symbol of $\mathbf{S}^{(b^*)}$ depend on the last two section symbols of $\mathbf{S}^{(\mathbf{p}^-)}$ and the first two section symbols of $\mathbf{S}^{(\mathbf{p}^+)}$ (similar to Equations 6.13-6.14).

From the VLMMs, we compute all possible section symbol sequences, $\left\{\boldsymbol{s_1^{(p^*)}}, \boldsymbol{s_1^{(p^*)}}, \ldots\right\}$, that obey the subsequence $\mathbf{S}^{(b^*)}$. From the possible section symbol sequences, we generate each path $\mathcal{P}^* = \left\{\mathbf{p}_1^*, \ldots, \mathbf{p}_{|\mathcal{P}^*|}^*\right\}$. The relative tempo of each node in the possible paths is set to the median of the relative tempo of all previously linked nodes, i.e. $\tau_R^{\left(\bar{s}_k^{(\mathbf{p}_r^*)}\right)} = \text{median}\left(\tau_R^{\left(\bar{s}_k^{(a)}\right)}, \forall \boldsymbol{\bar{s}_k^{(a)}} \in \boldsymbol{\mathcal{N}}(\boldsymbol{\mathcal{G}})\right)$, where $\bar{s}_k^{(\mathbf{p}_r^*)} \in \bar{\mathbf{S}}^{(\mathbf{p}_r^*)}$ (Figure 6.14a). Therefore the duration of the nodes in the path becomes $|t(\bar{s}_k^{(\mathbf{p}_r^*)})| = d(\bar{s}_n^{(b)})/\tau_R^{\left(\bar{s}_k^{(\mathbf{p}_r^*)}\right)}, \forall \bar{s}_k^{(\mathbf{p}_r^*)} \in \bar{\mathbf{S}}^{(\mathbf{p}_r^*)}$ and $s_k^{(\mathbf{p}_r^*)} = s_n^{(b)}$, where $\bar{s}_n^{(b)}$ is the section in the music score (b) with the identical label.

We then compare the duration of each path and the interval, $|t^* - t^*(\mathbf{p}_r^*)|$. We pick $\mathbf{p}_r^*$, such that $r = \text{argmin}_r\left(|t^* - t^*(\mathbf{p}_r^*)|\right)$ with the constraint $|t^* - t^*(\mathbf{p}_r^*)| < 3$ seconds. If no path is found, the interval is labeled as "unrelated" to composition, i.e. $s_k = \mathit{unrelated}$ (Figure 6.14b). Finally, all the links $\bar{s}_k^{(a)}$ are marked as section links.

6.7.3 Dataset

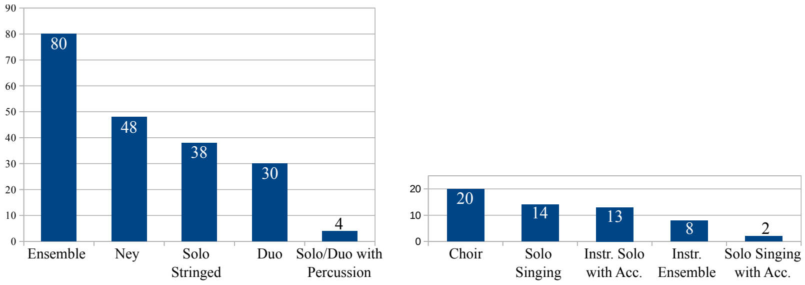
For the experiments, 200 audio recordings of 44 instrumental compositions (peşrev and sazsemaisis), and 57 audio recordings of 14 vocal compositions (şarkıs) are collected (i.e. 257 audio recordings of 58 compositions in total). The makam of each composition is included in the metadata.[17] The pieces cover 27 different makams.

The scores are taken from the **SymbTr** score collection (Karaosmanoğlu, 2012) (Section 3.1.2). The beginning and ending notes of each section are indicated in the instrumental **SymbTr**-scores. In the vocal compositions the sections can be obtained from the lyrics and the melody indicated in the **SymbTr**-score. In this Section, we manually label each section in the vocal compositions according to these. The section sequence indicated in the PDF formats is found in the **SymbTr**-txt and MIDI scores as well (i.e. following the lyric lines, the repetitions, volta brackets, coda signs etc. in the PDF). The duration of the notes in the MIDI and **SymbTr**-score are stored according to the tempo given in the PDF. We divided the MIDI files manually according to the section sequence given in the **SymbTr**-txt scores. MIDI files include the microtonal information in the form of pitch-bends.

Three peşrevs (associated with 13 recordings) do not have a Teslim section in the composition but each section has very similar endings (Section 2.1). Nine peşrevs (associated with 40 recordings) have less than 4 Hanes in the scores. There are notated tempo changes in the 4th Hanes of four sazsemaisi compositions (in the PDF), and the note durations in the related sections in the **SymbTr**-scores reflect these changes. In most of the şarkıs each line of the lyrics is repeated. Nevertheless, the repetition occasionally comes with a different melody, effectively forming two distinct sections. Two şarkı compositions include gazel sections (vocal improvisations). The mean and standard deviation of the duration of each section given in the score are 38.16 and 19.73 seconds for instrumental compositions, and 13.05 and 3.97 seconds for şarkıs.

The audio recordings are stored in MP3 format and the sampling rate is 44100 Hz. They are selected from the CompMusic corpus (Section 3.1.1) and they are either in public-domain or com-

[17] The metadata is stored in MusicBrainz: `http://musicbrainz.org/collection/5bfb724f-7e74-45fe-9beb-3e3bdb1a119e`



(a) **(b)**

Figure 6.15: Instrumentation and voicing in the reordings of **OTMM-section-linking a)** Instrumentation in the peşrevs and sazsemaisis **b)** Voicing in the şarkıs. "Instr." and "Acc." stands for instrumental and accompaniment, respectively.

mercially available. The ground truth is obtained by manually annotating the timings of all sections performed in the audio recordings. There are 1457 and 638 sections performed in the recordings of the instrumental and vocal compositions, respectively (a total of 2095 sections). In all the audio recordings, a section is repeated in succession at most twice. The mean and standard deviation of the duration of each section in the audio recordings are 35.17 and 19.49 seconds for instrumental, and 13.47 and 6.17 seconds for vocal pieces, respectively.

The performances contain tempo changes, varying frequency and kinds of embellishments, and inserted/omitted notes. There are also repeated or omitted phrases inside the sections in the audio recordings. Heterophonic interactions occur between instruments played in different octaves. Figure 6.15a,b shows the instrumentation and voicing of the audio recordings in the dataset. Among the audio recordings of instrumental compositions, ney recordings are monophonic. They are mostly from the "Instrumental Pieces Played with the Ney" collection (43 recordings),[18] and performed very similar to the score tempo and without phrase repetitions/omissions. From solo stringed recordings to ensemble recordings the density of heterophony typically increases. All audio recordings of vocal compositions are heterophonic. Hence the dataset represents both the monophonic and the heterophonic expressions in makam music. The ahenk (transposition) varies from recording to record-

[18] http://neyzen.com/ney_den_saz_eserleri.htm

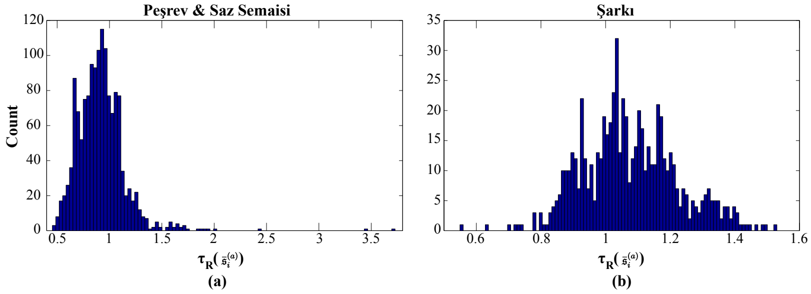


Figure 6.16: Histograms of relative tempo $\hat{\tau}$ in the dataset **a)** Peşrevs and sazsemaisis **b)** Şarkı. "Instr." and "Acc." stand for instrumental and accompaniment, respectively.

ing, which means that the tonic frequency (karar) varies even between interpretations of the same composition. Some of the recordings include material that is not related to any section in the score, such as taksims (non-metered improvisations), applauses, introductory speeches, silence and even other pieces of music. The number of segments labelled as $unrelated$ is 220.[19]

We computed the distribution of the relative tempo, which was obtained by dividing the durations of sections in a score by the duration of its occurance in a performance (Equation 6.4). Figure 6.16 shows all the occured quotients for the annotated sections in the audio recordings in the dataset. The outliers seen in Figure 6.16a are typically related to performances which omit part of a section, and 4th Hanes, which tend to deviate strongly from the annotated tempo. As can be seen from Figure 6.16, the tempo deviations are roughly Gaussian distributed, with a range of quotients $[0.5\ 1.5]$ covering almost all observations. This will help us to reduce the search space of our algorithm in Section 6.7.2.

6.7.4 Experiments

We link the sections given in the score with segments in the audio recordings using the approach described in Section 6.7.2. The signal features are either chosen as predominant melodies using SEN-MEL or HPCPs (Section 6.2). For comparison, the features are

[19] The score data, annotations and results are available in http://compmusic.upf.edu/node/171.

computed with 12, 24, 48 and 120 bins per octave (4.42, 2.21, 1.10, 0.44 Hc resolution). The binarization threshold $\beta(\boldsymbol{B})$ range from 0.5 to 9 Hc (*i.e.* a whole tone) for distance matrices calculated from predominant melodies, and from 0.20 to 0.50 for distance matrices computed from HPCPs.

Besides section linking, we extract pitch features from annotated audio segments and link them to the audio recording itself, i.e. "self-linking" the section annotations to the audio. This operation represents an upper limit for the possible results achievable by section linking (Figure 6.17). For repeated sections, the first annotation of any repeated section in the audio recording is selected for self-linking. Self-linking should ideally be able to link all the sections in the audio recording except the repeated sections with phrase omissions, repetitions and tempo changes.

We compare the initial and final times, t_{ini} and t_{fin}, of the sections after sequential linking and self linking with the manually annotated time intervals separately. A section is marked as a true positive, if an annotation in the audio recording and the link has the same section label, and the section is aligned with the annotation, allowing a tolerance of ± 3 seconds. All links that do not satisfy these two conditions are considered as false positives. If a section annotation does not have any links in the vicinity of ± 3 seconds, it is marked as false negative. If an unrelated section is aligned with an unrelated annotation, allowing a tolerance of ± 3 seconds, the unrelated section is a true negative. From these quantities we compute specificity, recall, precision, and F_1-scores as:

$$\begin{aligned} \text{Precision} &= \frac{t_p}{t_p + f_p}, \quad \text{Recall} = \frac{t_p}{t_p + f_n} \\ \text{Specificity} &= \frac{t_n}{t_n + f_p}, \quad F_1 = 2\,\frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \end{aligned} \tag{6.15}$$

t_p, t_n, f_n, f_p and F_1 stand for number of true positives, number of true negatives, number of false negatives, number of false positives and F_1-score, respectively. For all results below, the term "significant" has the following meaning: the claim is statistically significant at the $p = 0.01$ level as determined by a multiple comparison test using the Tukey-Kramer statistic.

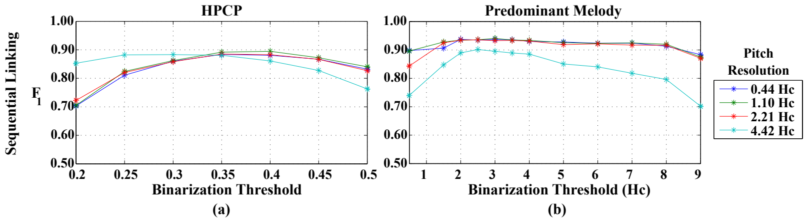


Figure 6.17: F_1 scores obtained for different pitch feature precision and binarization threshold. **a)** Sequential linking results using HPCPs and annotated karars, **b)** Sequential linking results using predominant melody and annotated karars.

6.7.5 Results

To find the optimal parameters for section linking, the experiments are done over a range of pitch feature precisions and binarization thresholds ($\beta(\boldsymbol{B})$) using annotated karars (Figure 6.17). The HPCPs with 4.42 Hc pitch precision (12 bins per octave) perform better with a binarization threshold at around 0.3. For pitch precisions higher than 4.42 Hc, the optimal results are obtained for a binarization threshold between 0.30 and 0.45 (Figure 6.17a). Increasing the precision produces slightly better but insignificant results ($p = 0.85$) for HPCPs. The optimal range of binarization threshold for predominant melody is observed between 1.5 and 4 Hc (Figure 6.17b). The F_1-scores are similar for HPCPs (88.3%) and predominant melody (0.90) with semi tone (4.42 Hc) pitch precision and optimal binarization thresholds at this precision ($\beta(\boldsymbol{B}) = 0.3$ for HPCPs and $\beta(\boldsymbol{B}) = 2.5$ Hc for predominant melodies). Since increasing the pitch precision more than quarter tone (< 2.21 Hc) does not have any significant effect, we select 2.21 Hc pitch precision as the optimal pitch precision. In the remainder of this section, we are reporting the detailed results using the optimal parameters (2.21 Hc pitch precision for both features; $\beta(\boldsymbol{B}) = 0.35$ for HPCPs and $\beta(\boldsymbol{B}) = 2.5$ Hc for predominant melodies), unless stated otherwise. The optimal parameters will be further discussed in Section 6.7.7.

The F_1-score obtained from the entire dataset using the optimal parameters and annotated karar is 0.89 (vs. 0.97 from self-linking) for HPCPs and 0.94 (vs. 0.97 from self-linking) for predominant melody, with the differences between the features being statisti-

Table 6.2: Section linking results obtained for candidate estimation and sequential linking with annotated karar, and for self-linking. Optimal parameters are used for computation. The results are given for the instrumental pieces and the vocal pieces separately.

		HPCPs			Predominant melody		
		Cand. Est.	**Seq. Link**	**Self Link**	**Cand. Est.**	**Seq. Link**	**Self Link**
Instr.	Precision	0.29	0.90	0.97	0.33	0.93	0.97
	Recall	0.89	0.86	0.97	0.94	0.92	0.97
	Specificity	0	0.47	0.78	0	0.55	0.73
	F_1	**0.44**	**0.88**	**0.97**	**0.49**	**0.92**	**0.97**
Vocal	Precision	0.43	0.92	0.98	0.54	0.97	0.96
	Recall	0.85	0.87	0.97	0.96	0.97	0.97
	Specificity	0	0.40	0.69	0	0.64	0.59
	F_1	**0.57**	**0.90**	**0.97**	**0.69**	**0.97**	**0.97**

cally significant. The average distance between each boundary of a true positive and the corresponding annotation is 0.36 and 0.44 seconds with a standard deviation of 0.41 and 0.49 seconds for links found from HPCPs and predominant melodies using optimal parameters, respectively. There is no significant change in F_1-scores with respect to the instrumentation of the instrumental pieces or voicing of the vocal pieces.

Table 6.2 gives the results for instrumental pieces and vocal pieces, using the annotated karar with optimal parameters. Apart from the results for the complete algorithm (Seq. Link), and the self-linking (Self Link), we also report the results obtained from the candidate estimation (Cand. Est.), i.e. without applying any candidate selection. While the precision is low for candidate estimation, sequential linking greatly increases the precision by effectively removing improbable candidates. In the meantime, the recall slightly drops for instrumental pieces, and a considerable number of non-linked intervals (44 t_p for HPCPs and 20 t_p for predominant melodies) are guessed correctly for vocal compositions, effectively increasing the recall. The specificity of candidate estimation is always 0, since unrelated time-intervals are marked in sequential linking (Section 6.7.2). Moreover, 155 (0.45 specificity) and 167 (0.57 specificity) unrelated annotations (out of 220 unrelated annotations) are correctly marked after guessing un-linked time intervals, using HPCPs and predominant melodies, respectively. While there is no significant difference between self-linking results for HPCPs and predominant melodies (0.97 F_1-score for HPCPs and 96.5% F_1-score for predominant melodies using optimal parameters), predominant melodies significantly outperform HPCPs in recall, precision and F_1-scores in both candidate estimation and sequential linking.

Using the predominant melody with optimal parameters and annotated karar, most of the errors are due to structural changes within sections in a performance (29 out of 129 false positives and 39 out of 146 false negatives in the whole dataset) and substantial tempo changes within the sections (23 out of 129 false positives and 30 out of 146 false negatives in the whole dataset) that Hough transform cannot handle. Most of these errors in the dataset arise from the 4^{th} Hane of the instrumental compositions, with 42 out of 129 false positives and 56 out of 146 false negatives in the instrumental compositions being related to the 4^{th} hane. glshough is not able to find the lines associated with 21 (out of 53 annotations) of the 4^{th} Hane annotations, due to their tempo deviating more than the allowed ratio with respect to the tempos given in the score. Remaining 4^{th} Hane annotations errors are due to omissions, repetitions and tempo changes observed in the performances. One audio recording is too slow[20] and two are too fast[21] such that the relative tempo of the sections are beyond the allowed interval. In these recordings Hough transform fails to detect the appropriate lines (2 true positives out of 24 section annotations, 14 false positives and 20 false negatives).

The results of section linking using automatic karar identification using the methodology explained in (Bozkurt, 2008) are reported in Table 6.3. Compared to section linking using annotated karar (Table 6.2), there is a considerable drop in all of the recall, pre-

[20] http://musicbrainz.org/recording/812828e6-3cb6-49c5-93fe-bf649c3096ae

[21] http://musicbrainz.org/recording/031a6e72-903a-479a-9a4c-e2a3335e4a0a, http://musicbrainz.org/recording/35d127d1-39e1-49d9-ab81-f8180b32590c

Table 6.3: Results obtained from automatic karar identification using optimal parameters for the instrumental pieces and the vocal pieces. The results are given for the instrumental pieces and the vocal pieces separately.

		HPCPs		**Predominant melody**	
		Cand. Est.	**Seq. Link**	**Cand. Est.**	**Seq. Link**
Instr.	Precision	0.22	0.76	0.23	0.78
	Recall	0.68	0.66	0.72	0.70
	Specificity	0	0.25	0	0.27
	F_1%	**0.33**	**0.71**	**0.35**	**0.74**
Vocal	Precision	0.35	0.85	0.45	0.88
	Recall	0.74	0.75	0.86	0.84
	Specificity	0	0.23	0	0.31
	F_1%	**0.48**	**0.80**	**0.59**	**0.86**

cision, and hence F_1-scores. This decrease is due to karar identification, which fails (i.e. the octave-wrapped distance between the annotated karar and identified karar are more than 2.5 Hc, i.e. the optimal binarization threshold) for 62 pieces (53 instrumental pieces and 9 şarkıs) out of 257 recordings (24.1% error). The karar identification typically fails in pieces with more complex makams such as *Ferahfeza* (10 out of 10 recordings), *Hicazkar* (8 out of 9 recordings), *Kürdilihicazkar* (9 out of 19 recordings), *Hüzzam* (5 out of 8 recordings), *Segah* (7 out of 14 recordings) and *Acemaşiran* (3 out of 7 recordings). For those makams, often more emphasis is put on notes different from the karar, which usually leads to the assignment of the karar to one of these notes. The F_1-score obtained from the entire dataset using the optimal parameters and automatic karar identification is 0.74% for HPCPs and 0.78% for predominant melody.

We also computed the elapsed time for section linking excluding feature computation and karar identification. On average, our implementation in MATLAB takes 3% of the duration of the audio recording (with a standard deviation of 1%) to link the sections of the particular audio recording with a 64 bit Ubuntu machine with

13.5 GB RAM and 3.33 GHz processor.

6.7.6 Section Linking Using Hierarchical Hidden Markov Models

Although highly accurate, the section linking methodology proposed in (Şentürk, Holzapfel, & Serra, 2014) requires manually annotated music scores and audio recordings to train parallel VLMMs for each unique section label sequence $\mathbf{S}^{(b)}$ of the studied compositions. Therefore, the sequential linking part of the section linking methodology is hard to scale to large music corpora of OTMM or other music cultures (Şentürk et al., 2016).

In (Holzapfel et al., 2015),[22] a simpler section linking method is proposed, which is based on a HHMMs. Unlike the bottom-up approach taken by the methodology in (Şentürk, Holzapfel, & Serra, 2014), the proposed method attempts to align the entire music score with the audio recording while allowing jumps in the score at the section boundaries. It should be noted that the constructed network assumes that the target audio performance does not include insertions (e.g. improvisations) with respect to the music score.

T is compared with the method proposed in (Şentürk, Holzapfel, & Serra, 2014) on a subset of 166 recordings (i.e. without şarkıs) from the **OTMM-section-linking** dataset. The recordings with extra contents are removed, following the proposed method's limitation to handle insertions unrelated to the music score. The evaluation measures in (Şentürk, Holzapfel, & Serra, 2014) (Section 6.7.4) is utilized for comparison. In addition the F_1-scores are computed for different temporal-tolerances (from 0.1 to 3 seconds) between the annotated and estimated section boundaries (Figure 6.18). In the experiments in a pairwise t-test is used for statistical significance computations at a $\alpha = 5\%$ significance level.

When the tolerance is selected as 3 seconds, the HHMM based method and the method proposed in (Şentürk, Holzapfel, & Serra, 2014) has an F_1 measure of 0.946 and 0.932, respectively. In this tolerance, there is no statitically significant difference betwen the methods. This shows that both methods give comparable results at

[22] I contributed to the publication by providing the dataset and reporting the results obtained by the method proposed in (Şentürk, Holzapfel, & Serra, 2014).

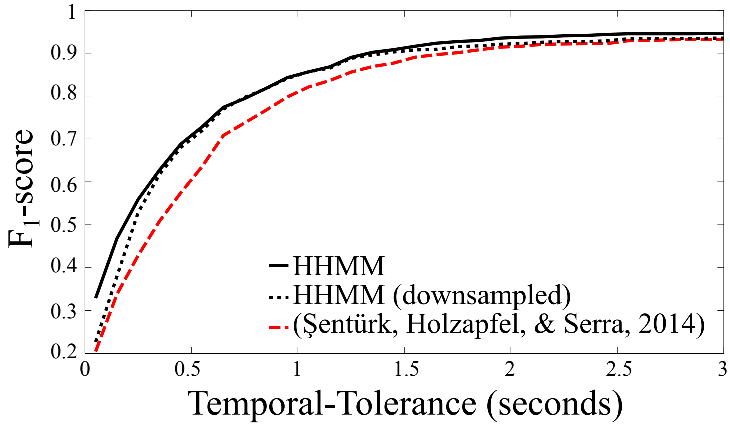


Figure 6.18: The F_1-scores for the HHMM based method, HHMM based method with downsampled predominant melody input and the method proposed in (Şentürk, Holzapfel, & Serra, 2014) for different temporal-tolerances.

locating the section boundaries. However the HHMM based method significantly outperforms the method proposed in (Şentürk, Holzapfel, & Serra, 2014) for temporal-tolerances less than or equal to 1 (0.846 vs 0.797 F_1-scores for 1 second tolerance). This is expected since the proposed model can adapt to intra-section tempo deviations, unlike the Hough transform based model.

Holzapfel et al. (2015) also compare the methods by reporting the average percentage of the audio duration for linking the sections. The implementation of the method proposed in (Şentürk, Holzapfel, & Serra, 2014) completes the operation in the 3% of the audio duration on average, while the implementation of the HHMM based method requires 25%. To improve the performance Holzapfel et al. (2015) downsamples the input predominant melody and synthetic melody by a factor of 3. This variant is able to retain comparable results within the 1.8% duration of the audio recording for tolerances longer than 300 miliseconds. This tolerance is highly sufficient, since it is close to the temporal-tolerances used in evaluating note-level aligment (200 ms is used in Section 6.8.3).

Nevertheless, the HHMM based linking method is not generalizable to large corpora either, since it is unable to align recordings with unrelated events, a common case in OTMM performances. In Section 6.12, several simplifications will be introduced to the section linking methodology proposed in (Şentürk, Holzapfel, & Serra, 2014) to be able to apply the joint analysis procedure automatically (described throughout this Chapter) to CompMusic OTMM corpus.

6.7.7 Discussion and Summary

The results show that our method is effective in linking sections given in the score to their corresponding time intervals in audio recordings, given a wide variety of instrumental and vocal timbres. The method is able to achieve good results by using different pitch features with fast computation time and accurate link boundaries.

To find optimal parameters for binarization threshold and pitch precision, we examined the results of section linking using different binarization threshold and pitch precision (Section 6.7.4). The optimal range for binarization threshold for predominant melody is between 1.5 and 4 Hc. Other than for HPCP, the threshold range of predominant melody has a musical interpretation since the performed notes might deviate from the theoretical frequency of a note by as much as a semi-tone, as explained in Section 2.1. This makes predominant melodies more intuitive to apply to OTMM, and possibly to other musics with a clear emphasis on melody.

For both HPCPs and predominant melodies, semitone pitch precision (4.42 Hc) performs worse than higher pitch precisions. This shows that pitch precisions higher than semitone are necessary to capture the melodic characteristics of OTMM. Nevertheless, increasing the pitch precision more than quarter tone (< 2.21 Hc) does not lead to further increase in the F_1-score. Therefore, both precision and binarization threshold lie in the vicinity of 2 Hc. While deviations between theory and practice were observed in single cases to reach a semi-tone, the usual deviation can be assumed to lie close to that value of 2 Hc. This proves the necessity of resolutions higher than the semi-tone resolution when attempting even such a high level task as we do in this paper. Even though increasing the pitch precision beyond 2.21 Hc (24 bins per octave) does not change the F_1-score practically, the default pitch precision can be increased further to use the same pitch tracks for more precision-demanding tasks such as karar identification, audio-score alignment or intonation analysis.

When comparing the results obtained from self-linking and sequential linking using predominant melody and annotated karar (Table 6.2), it is evident that the method is able to achieve practically the maximum possible F_1-score for vocal pieces (0.97 vs. 0.97 from self linking[23]) and a very high F_1-score for instrumental pieces (0.92 vs. 0.97 from self linking). The drop in the F_1-score for the instrumental pieces is mostly due to the errors related to specific performance characteristics (internal repetitions, omissions and tempo changes) which glshough cannot handle effectively. Moreover, most of these errors are related to the 4^{th} Hanes. In fact, resolving all the errors related to 4th Hanes would increase the F_1-score to approximately 0.96 (vs. 0.97 from self linking). In the sazsemaisi form, dividing the 4th Hane further into its substructures (Section 2.1) might help to handle these problems. More generally, such performance features could be better handled by aligning audio and the score at the note level.

Statistical significance tests show that our feature and similarity matrix computation is resilient to changes in timbre and density of heterophony. Moreover, recall obtained in candidate estimation step (Table 6.2) show that Hough transform is able to give reliable estimations for section links. It only fails in three audio recordings (out of 257) in which the performance is beyond the allowed tempo ratios. To remove the angle constraints from the line detection step, we need to estimate an average tempo of performance. Increasing the range of searched angles in Hough transform and then deducing the average tempo ratio of the performance from candidates with high weights might be sufficient for tempo estimation.

Comparing the recall of candidate estimation and sequential linking in Table 6.2, it can be seen that guessing non-linked intervals improves the section linking in vocal pieces. However it does not make any improvement in instrumental pieces, mostly due to the non-linearities in the performances of 4th hanes. Guessing non-linked intervals is dependent on the median of the relative tempo of the performed sections (see Section 6.7.2). In the case, where the 4th Hane does not follow the tempo indicated in the score or

[23] In both cases the number of true positives are the same (616 true positives), however section linking using annotated karar produces slightly less errors than self linking (20 vs. 23 false positives and 19 vs. 22 false negatives). This difference is insignificant.

there are structural deviations inside the performance of the section, the duration of the guessed paths do not match the duration of the non-linked audio segment.

The self-linking results (Table 6.2) imply that both chroma features and predominant melody can ideally perform equally well. However, candidate estimation using HPCPs misses more true positives than predominant melody, and sequential linking is not able to reduce the gap between the F_1-scores (Tables 6.2 and 6.3). This indicates that the predominant melody is a more adequate representation, when aiming at a comparison between score and audio in this musical context.

Our method can not search $unrelated$ annotations directly and the errors within the time interval can only be removed by Algorithm 4, for false positives that do not obey the section sequences. Moreover the unrelated region can not be marked correctly if the time interval of an unrelated region is estimated poorly due to a neighboring section with tempo changes, phrase repetition/omissions. Detecting "non-musical" events such as applause and silence can help to distinguish the unrelated regions and eliminate errors due to tempo changes typically occurring in the end of the recordings.

The bottleneck of the system is the automatic karar identification. If the karar of the piece is recognized incorrectly, no true lines will be present in the binarized similarity matrices obtained from either of the two feature types. While the results with automatic karar identification using Makam Toolbox are still good, the errors becomes a noticeable drawback especially for pieces composed in complex makams. Nevertheless, by using the melodic information in the scores we can greatly increase the accuracy of the karar identification accuracy. Recently, in (Şentürk et al., 2013), we extracted the stable pitches from the audio recording (i.e. the peaks of the pitch distribution computed from the prominent melody) and attempted to link the repetitive section in the score using the candidate estimation method explained in Section 6.7.2, assuming each stable pitch as the karar. Using the same data collection explained in Section 6.7.3 we achieved an accuracy of 99.6% (1 fail out of 257) effectively solving the karar identification problem for pieces with an available score. The F_1-scores obtained from section linking using repetitive section linking for karar identification is 0.89

(vs. 0.89 using annotated karar) for HPCPs and 0.93[24] (vs. 0.94 using annotated karar) for predominant melody for the whole dataset.

We presented a novel methodology to link musically relevant sections in a score with corresponding time intervals in an audio recording of the same piece. We tested our approach using HPCPs, a pitch feature previously applied to music with strong emphasis on harmony, and predominant melodies, a melodic pitch feature. We demonstrated that predominant melodies capture the heterophonic characteristics of OTMM better than HPCPs. Since scales in OTMM need resolutions higher than a semitone, we also tried section-linking over a range of pitch precisions and binarization thresholds. It was observed that the pitch precision has to be higher than semitone to represent the melodic granularity of OTMM. Unlike HPCPs, the optimal range of binarization threshold for predominant melodies was musically interpretable. Therefore using predominant melodies is more intuitive for music, where there is a clear melody, and concepts like functional harmony do not exist. Our results show the importance of culture-aware and knowledge-based systems. Nevertheless, we have also achieved remarkable results using HPCPs. It may be argued that that the methodology can be easily adapted to Eurogenetic musics, which can be typically conceptualized with the help of harmony.

Our approach is fast and accurate in matching both the section labels and their corresponding time intervals. Section links can be used as a complementary information in computational tasks such as form analysis and audio score alignment. Moreover, the computational steps in our approach can be modified to be used in similar research problems such as pattern matching and version detection. In (Şentürk et al., 2013), we used the candidate estimation methodology (Section 6.3) to identify the performed karar of the audio recording. Our results indicate that score information greatly simplifies the karar identification task. Parallel to findings of Aucouturier and Sandler (2002), the self-linking results (Fig-

[24] Using predominant melody with optimal parameters, all of the sections are correctly linked (i.e. 1 recall) in the audio recording with failed karar identification[25]. In this recording the distance between the estimated and actual karar is slightly higher than 2.5 Hc, i.e. the optimal binarization threshold. Our section linking methodology is still able to link the sections even though the resultant weights are low.

ure 6.17c,f) imply that repetitive section linking can be effectively used for audio thumbnailing.

Nevertheless, there is still room for improvement for a more reliable automatic system. As a next step in this research we want to increase the granularity of the linking between audio and score in order to provide more insights on the dynamics and the intonation of makam music performances. This should solve section linking problems due to performance particularities such as omissions, repetitions and tempo changes. We plan to use JumpDTW proposed by Fremerey et al. (2010), which allows jumps between the measures indicated in the score. Since we know the score section sequence, we can further modify JumpDTW to allow jumps between the sections and between the measures within each section. Section linking prior to audio-score alignment might increase the F_1-score and reduce computational time.

Content-based creation, analysis and discovery of multimodal and inter-linked music collections is becoming an active research area in the last few years, thanks to the advances in information technology and the emergence of vast numbers of available multimodal information sources such as audio, glsMIDI, sheet music, video and editorial metadata (Cornelis, Lesaffre, Moelants, & Leman, 2010; Thomas et al., 2012). Under the CompMusic Project we are developing *Dunya*, a system to browse and interact with music collections in a culturally informed way (Sordo, Koduri, Şentürk, Gulati, & Serra, 2012; Porter et al., 2013a). We will integrate our *section linking* methodology to the system and use it as a culture-specific tool for navigation and discovery of OTMM. We hope that our approach will contribute to the information technologies aimed at preserving, discovering and appreciating musical cultures.

6.8 Note-Level Audio-Score Alignment

In this Section, we extend the alignment methodologies explained so far to cover note-level alignment.

This remainder of the Section is structured as follows: Section 6.8.1 explains the note-level alignment. Section 6.8.2 presents the data collection. Section 6.8.3 presents the experiments. Section 6.8.4 presents the results and Section 6.8.5 give a brief discussion and conclusion.

6.8.1 Methodology

Given a music score of a composition with structure (section) information and an audio performance of the same composition, our method first extracts a synthetic melody $\hat{\boldsymbol{\Psi}}^{(\bar{s}_j^{(b)})}$ per section $(\bar{s}_j^{(b)})$ from the note values and durations in the score (b) and a audio predominant melody $\boldsymbol{\varrho}^{(a)}$ from the audio recording (a) (Section 6.2). Then it identifies the performed tonic frequency $\kappa^{(a)}$ using the score-informed tonic identification method explained in (Şentürk et al., 2013) (Section 6.4). The audio predominant melody is normalized with respect to the estimated tonic and $\hat{\boldsymbol{\varrho}}^{\kappa^{(a)},(a)}$ is obtained.

Next each score section $(\bar{s}_j^{(b)})$ is linked with the time intervals $t(\bar{s}_i^{(a)}), \forall s_i^{(a)} = s_j^{(b)}$, where each score section is performed in the audio recording (a) (i.e. structure level alignment) (Section 6.7).

After section linking, the obtained time-interval of each audio section is extended by 3 seconds to deal with the tempo differences. For each section link $\pi(\bar{s}_i^{(a)}, \bar{s}_j^{(b)}), s_i^{(a)} = s_j^{(b)}$, the synthetic melody is recomputed by resampling the feature according to the estimated tempo of the recording and the synthetic melodies $\hat{\boldsymbol{\Psi}}^{\tau^{(a)},(\bar{s}_j^{(b)})}$ are obtained. Next, subsequence dynamic time warping (Müller, 2007, Chapter 4) is applied between the normalized audio predominant melody $\hat{\boldsymbol{\varrho}}^{\kappa^{(a)},(s_i^{(a)})}$ of each section in the audio recording and synthetic melody $\hat{\boldsymbol{\Psi}}^{\tau^{(a)},(\bar{s}_j^{(b)})}$ of the corresponding section in score. We use the parameters as described in Section 6.3.2 (Figure 6.19a). In addition, we also apply global constraint as discussed in (Sakoe & Chiba, 1978). The bandwidth of the global constraint is selected as 20% of the query length. As a result the note sequence $\bar{\mathbf{N}}^{\left(\bar{s}_i^{(a)}\right)} = \left[\bar{n}_1^{\left(\bar{s}_i^{(a)}\right)}, \ldots, \bar{n}_{\left|\bar{\mathbf{N}}^{\left(\bar{s}_i^{(a)}\right)}\right|}^{\left(\bar{s}_i^{(a)}\right)}\right]$ of each section $\bar{s}_i^{(a)}$ is obtained. Note that note-level alignment refines the time interval of the section $t\left(\bar{s}_i^{(a)}\right) = \left[t_{ini}\left(\bar{n}_1^{\left(\bar{s}_i^{(a)}\right)}\right)\ t_{fin}\left(\bar{n}_{|\bar{\mathbf{N}}^{\left(\bar{s}_i^{(a)}\right)}|}^{\left(\bar{s}_i^{(a)}\right)}\right)\right]$ and the align-

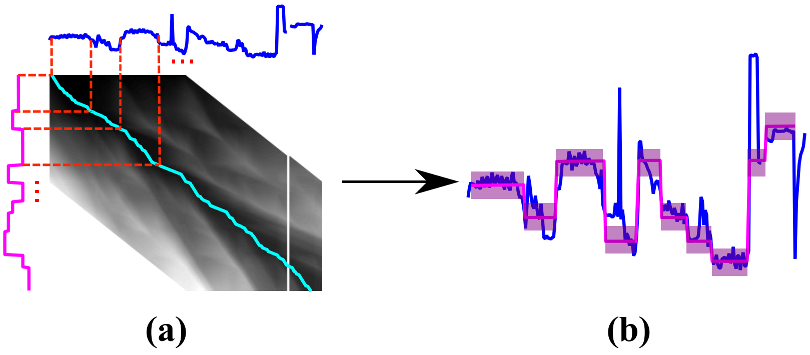


Figure 6.19: Note-level audio score alignment using SDTW. **a)** The resultant alignment path displayed on top of the accumulated cost matrix between the score synthetic pitch and audio predominant melody, **b)** Notes inferred from the alignment path

SymbTr-score	Audio MBID	Instrumentation	#Anno	t_p	f_p	f_n	F_1%
beyati–pesrev–hafif—-seyfettin_osmanoglu	70a235be-074d-4b9b-8f94-b1860d7be887	ensemble	906	790	116	116	87.2
huseyni–pesrev–muhammes—-lavtaci_andon	8b78115d-f7c1-4eb1-8da0-5edc564f1db3	ensemble	614	482	132	132	78.5
	9442e4cf-0cb3-4cb3-a060-77aa37392501	ney & percussion	302	260	45	42	85.7
rast–pesrev–devrikebir—-giriftzen_asim_bey	31bf3d56-03d8-484e-b63c-ae5ae9a6e733	tanbur	658	374	306	281	56.0
	5c14ad3d-a97a-4e04-99b6-bf27f842f909	ney	673	418	262	255	61.8
segah–pesrev–devrikebir—-yusuf_pasa	e49f33b8-cf8a-4ca9-88cf-9a994dbad1c0	ney & kanun	743	267	490	476	35.6

Table 6.4: Results of note-level alignment per experiment.

ment path $\varpi\left(\bar{s}_i^{(a)}, \bar{s}_j^{(b)}\right)$ such that $s_i^{(a)} = s_j^{(b)}$ (Figure 6.19b).

6.8.2 Dataset

For the experiments, the scores for each composition are obtained from the **SymbTr** collection (Karaosmanoğlu, 2012). 6 audio recordings of 4 peşrev compositions are selected from the automatic transcription dataset presented in (Benetos & Holzapfel, 2013). The recordings are performed in a variety of transpositions. There are 51 sections in the audio recordings in total. The duration of the sections are 36.1 seconds on average with a standard deviation of 16.2 seconds.The total number of the note annotations in the audio recordings are 3896. These annotations typically follow the note sequence in the **SymbTr**. Note that there are 3 inserted and 49 omitted notes in the annotations with respect to the **SymbTr**-scores.

6.8.3 Experiments

Given a score of a composition and an audio recording of the same composition, we align the notes in the **SymbTr**-score of a composition with the corresponding audio performance of the same composition using the methodology explained in Section 6.8.1 and obtain aligned note onsets in the audio recording.

To evaluate the tonic identification, we compare the distance between the pitch class of the estimated tonic and the pitch class of the annotated tonic as explained in (Şentürk et al., 2013). If the distance is less than 1 Hc, the estimation is marked as correct.

To evaluate section linking, we check the time distance between the time interval of annotated sections and sections links as explained in Section 6.7.4 (Şentürk, Holzapfel, & Serra, 2014). A section link is marked as a true positive, if an annotation in the audio recording and the link has the same section label, and the link is aligned with the annotation, allowing a tolerance of ± 3 seconds. All links that do not satisfy these two conditions are considered as false positives. If a section annotation does not have any links in the vicinity of ± 3 seconds, it is marked as false negative.

To evaluate the note-level alignment, we compare the aligned onset $\bar{n}_k^{\left(\bar{s}_i^{(a)}\right)}$ and the corresponding annotated onset $\bar{\mathfrak{n}}_k^{\left(\bar{\mathfrak{s}}_i^{(a)}\right)}$ ($n_k^{\left(\bar{s}_i^{(a)}\right)} = \mathfrak{n}_k^{\left(\bar{s}_i^{(a)}\right)}$). We consider the aligned onset $t_{ini}\left(\bar{n}_k^{\left(\bar{s}_i^{(a)}\right)}\right)$ as a true positive if its time-distance to the annotated onset $t_{ini}\left(\bar{\mathfrak{n}}_k^{\left(\bar{\mathfrak{s}}_i^{(a)}\right)}\right)$ is less than ± 200 ms. If the time-distance is higher than ± 200 ms, the aligned onset and the annotated onset are labeled as false positive and false negative, respectively. The insertions are ignored in the evaluation. If the aligned note corresponding to an omitted annotation is not rejected (i.e. the duration $t\left(\bar{n}_k^{\left(\bar{s}_i^{(a)}\right)}\right) > 0$), it is deemed as a false positive.

From these quantities we compute the F_1-scores for section linking and note-level alignment separately Equation 6.15.

6.8.4 Results

Across all the experiments, the tonic is identified correctly (100% accuracy in tonic identification) and all the sections were linked perfectly ($F_1 = 100\%$ for section linking). In the note-level, our methodology is able to align 2591 notes out of 3896 notes correctly, yielding to an F_1-score of 0.66. The mean, median and standard deviation of the time-distance between the aligned note and the corresponding annotation are 299, 93 and 498 milliseconds, respectively. Moreover, 89% (recall rate) of the notes are aligned with a margin of ±1 second, implying that SDTW does not lose track of the melody.

Previously in (Şentürk et al., 2013) and (Şentürk, Holzapfel, & Serra, 2014) we showed that our linking methodology is highly reliable for tonic identification and section linking. The results in this paper also comply with these previous findings.

To understand the common mistakes in the note-level, we examined the aligned notes against annotated notes. Table 6.4 shows the results per experiment. The expressive embellishments in the performance (*portamento*s, *legato*s, *trill*s etc.) are common reasons of misalignment. For example, SDTW infers portamentos as an insertion and the note onsets are aligned around the time when the portamento reaches to the stable note pitch. Similarly when there is a melodic interval less than a whole tone, a trill might cause a note onset to be marked earlier. Since these embellishments are not shown in the score, standard (subsequence) SDTW was expected to fail. While we can argue that the section-level alignment is accurate, the results in the note-level alignment show that there is still more room for improvement for note-level alignment.

From Table 6.4, it can be seen that the note-level alignment fails for most of the notes in the audio recording of *Segah Peşrev* within the ±200ms tolerance. This is a recording with ney and kanun, which consists of heterophonic interactions such as embellishments played by a single musician and time differences in note onsets between the performers. Due to such cases, the time distance between aligned onset and the annotated is typically larger than 200ms. Note that 75% (recall rate) of the notes are still aligned correctly within a tolerance of ±1 second.

6.8.5 Discussion and Summary

In this Section, we proposed a method to align scores of makam musics with their associated audio recordings. Our system is able to handle the transpositions and structural repetitions and omissions in the audio recordings, which are common phenomenon in makam musics. The results obtained from the data collection present a proof-of-concept that a standard technique such as SDTW can be effective for audio-score alignment for makam musics in the note level. Nevertheless, we need incorporate additional steps to handle non-notated embellishments and note omissions, insertions and repetitions.

Currently method relies on manual section segmentations in music scores. Manual segmentation of the score is not an difficult task compared to the note-level audio-score alignment itself. Nevertheless, it might be desirable to use other methodologies that do not require structural segmentations (e.g. (Grachten et al., 2013)), especially when we are working on large audio-score collections.

While we didn't have such an example in our data collection, there can be also omissions, insertions and repetition of phrases inside the sections. Currently, our methodology cannot handle such cases. In the future we want to use the JumpDTW proposed by (Fremerey et al., 2010) to handle these intra-section omissions, insertions and repetitions. Another approach might be segmentation of the symbolic score into melodic phrases and link extracted phrases from score with the corresponding audio recording. Recently, Bozkurt et al. (Bozkurt, Karaosmanoğlu, et al., 2014) came with a method for segmenting music scores into melodic phrases according to the makam and usual information. Our initial experiments using the extracted phrases show that phrase linking is highly accurate. We observed that the erroneously linked phrases are almost identical to the true phrase, differing by very few pitches or durations, hence note-level alignment does not suffer a large number of errors.

We are extending the data collection to cover more examples from the CompMusic collection. In audio recordings with heterophonic interactions (such as the audio recording of *Segah Peşrev*) there is an ambiguity of the exact timings in the note onsets. To study the implications we plan to make several experts to annotate the note onsets in the same set of scores and audio recordings. We will jointly compare the onset markings from each annotator with the aligned onsets produced by the future iterations of our automatic audio-score alignment method.

6.9 Inferring Measures, Phrases, Lyrics, Onsets and Usul Strokes

Up to here, the note sequence $\bar{\mathbf{N}}^{(a)} = \left[\bar{n}_1^{(a)}, \bar{n}_2^{(a)}, \ldots,\right]$ in the audio recording (a) is obtained. The notes in $\bar{\mathbf{N}}^{(a)}$ are linked with the notes in the note sequence $\bar{\mathbf{N}}^{(b)} = \left[\bar{n}_1^{(b)}, \bar{n}_2^{(b)}, \ldots,\right]$ of the score (b). Remember that the **SymbTr**-scores contain the measure sequence $\bar{\mathbf{M}} = \left[\bar{m}_1^{(b)}, \bar{m}_2^{(b)}, \ldots,\right]$ and the lyrics $\boldsymbol{\lambda}^{\left(\bar{n}_k^{(a)}\right)}, \forall \bar{n}_k^{(a)} \in \bar{\mathbf{N}}^{(a)}$ are coupled with the notes in the syllable-level (Section 3.1.2). If automatic phrase segmentation has been applied to the score (Section 4.3), a phrase sequence $\bar{\mathbf{P}}^{(b)} = \left[\bar{p}_1^{(b)}, \bar{p}_2^{(b)}, \ldots,\right]$ is also obtained for the music score. The measures, phrases and lyrics in the music score can be linked with the respective events in the audio recording by simply referring to the alignment between the note sequences $\bar{\mathbf{N}}^{(a)}$ and $\bar{\mathbf{N}}^{(b)}$, and the start and final note indices of each event in the score. Note that the melody of a section may be identical or very similar to the melody of other sections, while the lyrics are different (Section 4.3.2). In such cases incorporation of automatic audio-to-lyrics alignment (Dzhambazov et al., 2016) is necessary to properly match of the lyrics.

Rhythmic information can be also easily inferred from the alignment between the audio recordng and the score. The onsets are trivially given as $t_{ini}\left(\bar{n}_k^{\left(\bar{s}_i^{(a)}\right)}\right)$. Note that these onsets does not constitute all the onsets in the audio recording *per se*, as the percussive onsets, note insertions (with respect to the music score) and heterophonic interactions related to the same note event are not notated in the score. Remember that the usul information parsed from the **SymbTr**-scores includes the usul changes throughout the piece (Section 4.1). By referring to an $\langle usul, stroke\_sequence\rangle$ dictionary, the usul strokes (e.g. the onomatopoeic stroke names such as düm and tek) and the symbolic duration of each stroke (e.g. ♪, ♩) may be obtained for each usul cycle. These strokes may be mapped to the note indices in the score and then to the sample indices in the audio recording (a) by referring to each alignment path $\boldsymbol{\varpi}\left(\bar{s}_i^{(a)}, \bar{s}_j^{(b)}\right)$ for each $\bar{s}_i^{(a)}$ such that $s_i^{(a)} = s_j^{(b)}$.

6.10 Score-Informed Predominant Melody Correction

The octave errors in the predominant melody are corrected by referring to the aligned note sequence $\bar{\mathbf{N}}^{(a)} = \left[\bar{n}_1^{(a)}, \bar{n}_2^{(a)}, \ldots,\right]$ in the audio recording (a). First, a modified audio note sequence $\bar{\mathbf{N}}^{*(a)}$ is obtained by carrying the end time $t_{fin}(\bar{n}_k^{(a)})$ of each note within the note sequence $\bar{\mathbf{N}}^{(a)}$ to min $\left(t_{fin}(\bar{n}_k^{(a)}) + 3 \text{ seconds}, t_{ini}(\bar{n}_{k+1}^{(a)})\right)$, so that all pitch samples will be covered within the aligned sections. Next, the modified note sequence is synthesized ($\hat{\boldsymbol{\Psi}}^{*(a)}$) according to the AEU theory theory (Section 4.2.2). Then the octave of each audio pitch sample $\rho_i^{\kappa^{(a)},(a)}$ in the audio normalized predominant melody $\boldsymbol{\varrho}^{(a)}$ is moved such that the absolute cent-distance to the aligned synthetic pitch $\hat{\psi}_j^{(a)}$ in the synthetic melody $\hat{\boldsymbol{\Psi}}^{*(a)}$ is minimized.[26] Figure 6.20 shows the octave-correction applied on a short except of a performance[27] of *Uşşak Sazsemaisi*.[28]

In the joint analysis methodology (Section 6.12), the input predominant melody is computed by `ATL-MEL` and then octave-correction is applied using the note sequence obtained from the alignment of each relevant music score. This procedure is termed as `SEN-MEL`$_f$. Using the octave-corrected predominant melody, the PD, PCD and melodic progression features are also recomputed.

[26] The implementation is available at `https://github.com/sertansenturk/alignedpitchfilter`.

[27] `http://musicbrainz.org/recording/e72db0ad-2ed9-467b-88ae-1f91edcd2c59`

[28] `http://musicbrainz.org/work/ad9fb46e-eb95-4446-93ad-e3bf13c01a95`. The score is online at `https://github.com/MTG/SymbTr/blob/v2.4.2/txt/ussak--sazsemaisi--aksaksemai----dede_salih_efendi.txt`.

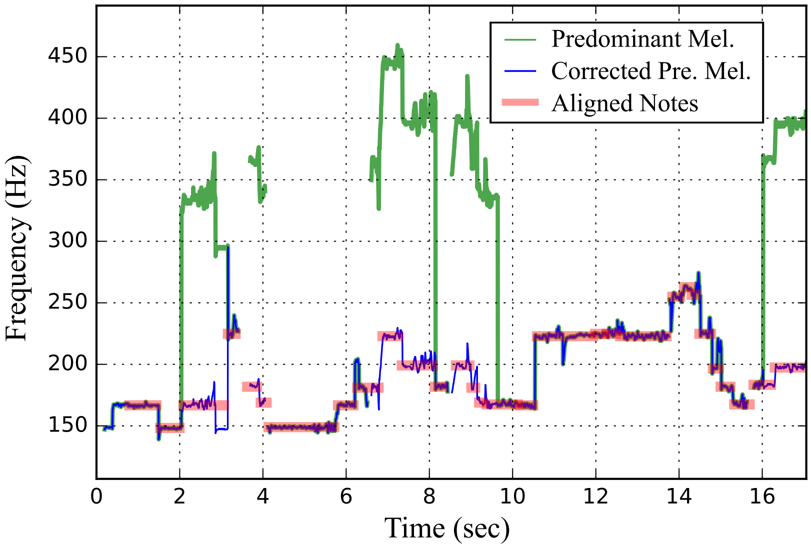


Figure 6.20: Octave correction in a short except of *Uşşak Sazsemaisi*. The initial predominant melody computed by `SEN-MEL`, the octave-corrected predominant melody and the aligned notes are shown as green, blue and red lines, respectively.

6.11 Score-Informed Tuning and Intonation Analysis

A musical note can be defined as a sound with a definite pitch and a given duration. An interval is a difference between any two given pitches. Most melodic music traditions can be characterized with a set of notes it uses and the corresponding intervals. They constitute the core subject matter of research concerning the tonality and melodies of a music system. For any quantitative analyses therein, it is required to have a working definition and a consequent computational model of notes which dictate how and what we understand of the pitch content in a music recording.

In much of the research in music analysis and information retrieval, the most commonly encountered model is one that considers notes as a sequence of points separated by certain intervals on frequency spectrum. There are different representations of the pitch content from a given recording based on this notion, the choice among which is influenced to a great degree by the intended application. Examples include pitch class profiles (Fujishima, 1999), harmonic pitch class profiles (Gómez, 2006) (Section 5.3), pitch distribution (Gedik & Bozkurt, 2010) and pitch-class distribution (Chordia & Şentürk, 2013) (Section 5.5) besides others. Albeit a useful model of notes used alongside several information retrieval tasks, we believe it is limited in its purview. In this Section, we discuss a score-informed tuning and intonation analysis method, consisting of a statistical model of notes (namely PDs each note symbol) that broadens the scope of the former, encapsulating the notion of the variability in notes.

This analysis methodology is originally proposed for Carnatic music in (Şentürk et al., 2016). The original methodology uses a variant of the section and note-level audio-score alignment steps explained throughout this Chapter, which is adapted and optimized for the culture-specific properties of Carnatic music. The methodology is evaluated extrinsically in a classification task comparing the results with a state-of-the-art system (Koduri et al., 2014) on two datasets of Carnatic music. The readers are referred to Appendix B.1 for the detailed description of the original methodology and the experiments.

Referring to the time intervals of each aligned note ($t(\bar{n}_i^{(a)}$), we extract the pitch trajectories $\boldsymbol{\varrho}^{(\bar{n}_i^{(a)})}$ of each note from the octave-corrected predominant melody. The median of each pitch trajectory is assigned as the the performed stable pitch $\phi^{\left(\bar{n}_i^{(a)}\right)}$ of the note. Next, the trajectories are grouped with respect to the note symbols, (e.g. the set of pitch trajectories for the note gerdaniye is $\left\{\boldsymbol{\varrho}^{\left(\bar{n}_i^{(a)}\right)} : n_i^{(a)} = \text{“gerdaniye”}\right\}$. The pitch values are aggregated from the set of pitch trajectories for each note symbol and they are used to compute a PD for each note symbol (Koduri et al., 2014) (e.g. $\boldsymbol{H}_P^{(a,\text{“gerdaniye”})}$). The peak in the PD of each note, which is closest to the theoretical tuning frequency is taken as the performed stable pitch of the note symbol (e.g. $\phi^{(a,\text{“gerdaniye”})}$). These features provide a means to specify both the overall and individual intonation and tuning characteristics of the performed notes.

In this step, the identified tonic pitch is also recomputed as the stable pitch of the tonic symbol $\kappa^{(b)}$ as indicated in the score (b), i.e. $\kappa^{(a)} = \phi^{\left(a,\kappa^{(b)}\right)}$. Notice that the tonic octave is also identified in

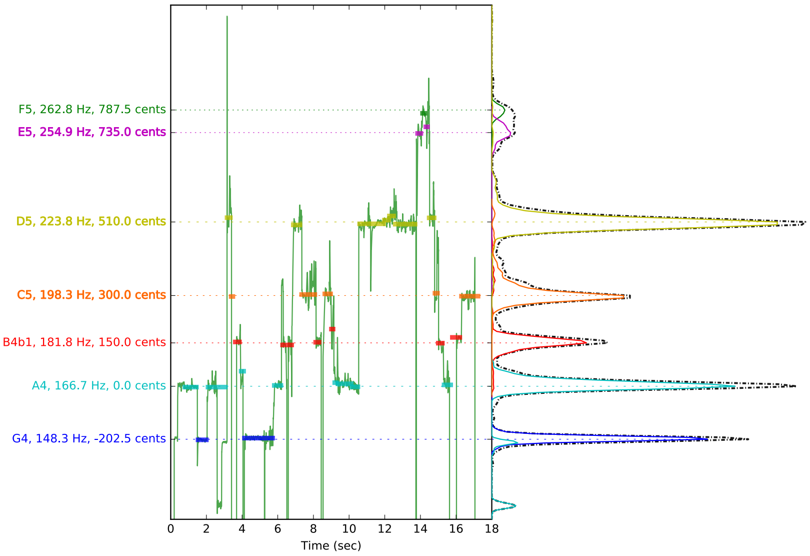


Figure 6.21: Score-informed tuning and intonation analysis applied on the octave-corrected predominant melody of the same short except of *Uşşak Sazsemaisi*, presented in Figure 6.20. The pitch distribution is drawn as a black-dashed curve. The note PDs are drawn in different colors with the note symbol, its frequency and cent-distance to tonic indicated on the left.

the process, whereas the identification methods output pitch-class of the tonic.[29] The transposition is also re-identified at this step from the refined tonic frequency (Section 5.8).

6.12 Combining Joint Audio-Score Analysis Methodologies

To obtain the automatic description of the audio recordings and music scores in the CompMusic makam corpus, the methodologies explained throughout this Chapter are implemented with several adjustments based on the experimental findings. Some of the algorithms are also simplified for the sake of scalability. Figure 6.22

[29] The implementation is available at https://github.com/sertansenturk/alignednotemodel.

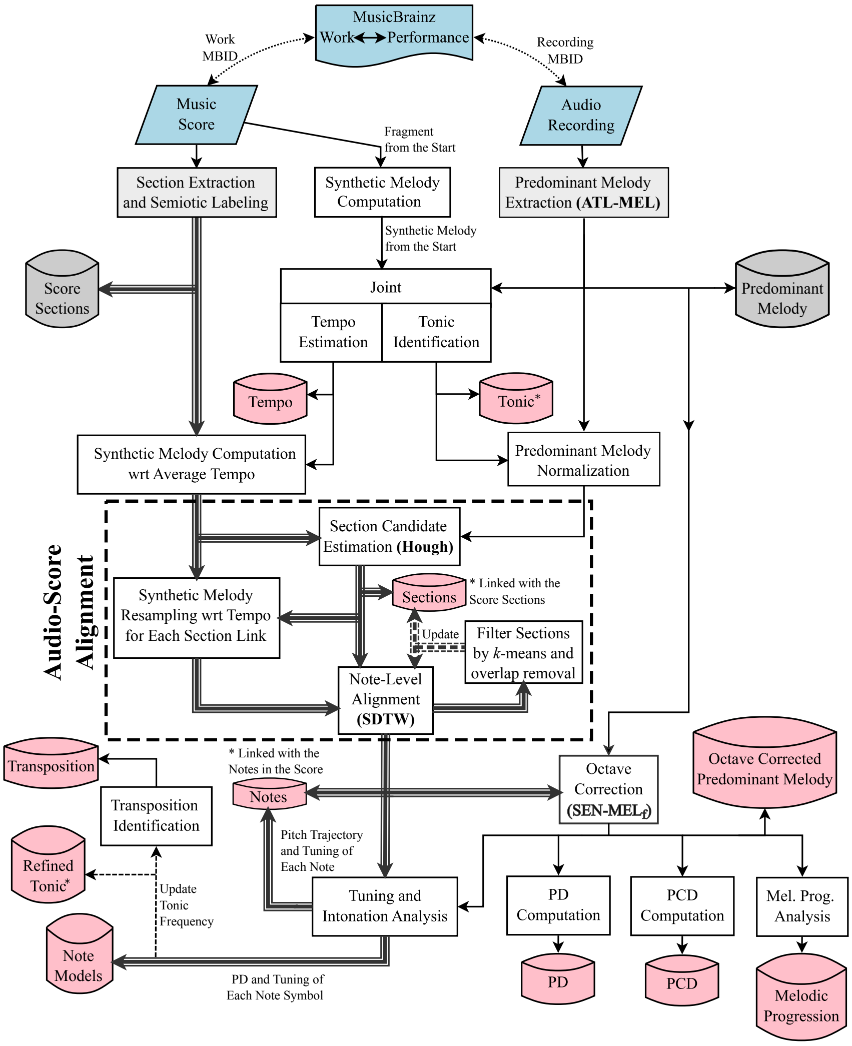

MusicBrainz
Work
Performance
Work MBID
Recording MBID
Music Score
Audio Recording
Fragment from the Start
Section Extraction and Semiotic Labeling
Synthetic Melody Computation
Predominant Melody Extraction (ATL-MEL)
Synthetic Melody from the Start
Score Sections
Joint
Tempo Estimation
Tonic Identification
Predominant Melody
Tempo
Tonic*
Synthetic Melody Computation wrt Average Tempo
Predominant Melody Normalization
Audio-Score Alignment
Section Candidate Estimation (Hough)
Sections
* Linked with the Score Sections
Synthetic Melody Resampling wrt Tempo for Each Section Link
Update
Filter Sections by k-means and overlap removal
Note-Level Alignment (SDTW)
Octave Corrected Predominant Melody
Transposition
* Linked with the Notes in the Score
Notes
Octave Correction (SEN-MEL$_{F}$)
Transposition Identification
Pitch Trajectory and Tuning of Each Note
Refined Tonic*
Update Tonic Frequency
Tuning and Intonation Analysis
PD Computation
PCD Computation
Mel. Prog. Analysis
Note Models
PD and Tuning of Each Note Symbol
PD
PCD
Melodic Progression

Figure 6.22: Joint Audio and Score Analysis Process. The process is repeated for each score of the composition performed in the audio recording

shows the block diagram of the joint audio-score analysis methodology. Below the overall methodology is explained:

Preliminary Score Analysis: The sections $\bar{s}_j^{(b)}$ in the score (b) are extracted using the implicit information in the music score (Section 4.3.2) and labeled semiotically (Section 4.3.2).

Preliminary Audio Analysis: The predominant melody $\boldsymbol{\varrho}^{(a)}$ of the audio recording (a) is extracted using ATL-MEL instead of SEN-

-MEL.

Joint Tonic Identification and Tempo Estimation: Based on the results obtained in the composition identification experiments (Table 6.1), a synthetic pitch $\hat{\mathbf{\Psi}}^{\left(\bar{f}^{(b)}\right)}$ is computed from a 15 second fragment. The fragment is selected from the start of the score in instrumental compositions and the start of the first vocal section in vocal compositions.[30] Hough transform is used at the fragment linking step and max similarity (Equation 6.10) is assigned to the weight of each tonic candidate.[31]

The average tempo of the audio recording $\tau^{(a)}$ is jointly estimated as the tempo of the audio fragment with the highest similarity obtained during the tonic identification step, as explained in Section 6.5.

Audio-Score Alignment: As mentioned earlier in Section 6.7.6, the section linking methodology proposed in (Şentürk, Holzapfel, & Serra, 2014) is not scalable to large corpora. For this reason, I introduced several simplifications and modifications to the original methodology. Moreover note-level alignment is incorporated into the section linking process for the sake of simplicity.[32]

1. In (Şentürk, Holzapfel, & Serra, 2014), the sections are labeled manually according to their melody. Instead, the automatically extracted sections with unique melodic labels are used (Section 4.3.2).

2. In (Şentürk, Holzapfel, & Serra, 2014), only the first occurrence of each section label is used, ignoring other variants (e.g. the instances of a repetitions with volta brackets). Here, the sections with "unique" melodies (i.e. "unique" cliques in the graph computed from the melodic dissimilarities between the synthetic pitch of each section, as explained in Sec-

[30] Remember the instrumental intros may be skipped or replaced by another instrumental intro in the vocal compositions 2.1.

[31] The compiled binary for joint tonic identification and tempo estimation is available at `https://github.com/sertansenturk/tomato_binaries` with the wrappers available in `tomato` (Appendix C).

[32] The compiled binary for section linking and note-level alignment is available at `https://github.com/sertansenturk/tomato_binaries` with the wrappers available in `tomato` (Appendix C).

tion 4.3) are selected, so that the variants of the same melody are also considered in the alignment.

3. In (Şentürk, Holzapfel, & Serra, 2014), the synthetic melody $\hat{\boldsymbol{\Psi}}^{\left(\bar{s}_j^{(b)}\right)}$ of each section is computed with respect to the nominal tempo $\tau^{(b)}$ in the score (Section 4.2.2). Here, the estimated average tempo $\tau^{(a)}$ of the audio recording is used.

4. Following Hough transform, SDTW is applied the between score section $\bar{s}_j^{(b)}$ and the audio section candidate $\bar{s}_i^{(b)}$ on the path of each detected line segment $\boldsymbol{\varpi}\left(\bar{s}_i^{(a)}, \bar{s}_j^{(b)}\right)$. This way not only the note-level alignment is obtained (Section 6.8), but the time-boundaries of the sections are also marked more accurately.

5. Motivated by the performance of the irrelevant document rejection step in composition identification (Section 6.6.2), the section candidates are initially (before sequential linking, explained in Section 6.7.2) clustered into two groups, and the candidates in the group constituting lower similarities are removed. The similarity-values computed for each audio section candidate are first normalized by normalizing the values such that the values have a zero mean and a standard deviation of one. For the sake of simplicity, k-means clustering, an unsupervised clustering method (Arthur & Vassilvitskii, 2007), is applied to the normalized similarity values using the squared Euclidean distance. This step greatly reduces the false positives without the need of sophisticated models such as VLMMs as described in Section 6.7.2.

6. The method in (Şentürk, Holzapfel, & Serra, 2014) requires training a VLMM for each section sequence in different forms. The VLMMs used in the path computation is removed (Equation 6.11) and the transition from one section to a connected section (boundary distance is less than 3 seconds) is allowed without any penalty. Next, the overlapping sections are removed (Algorithm 3). Inconsequent section removal and guessing unsure time-intervals (described in Section 6.7.2) are also skipped as they too require VLMMs.

The methodologies used in the joint analysis are implemented in Python, and they are integrated to tomato[33] The algorithms are designed such that partial or complete fails (due to missing information) at each step are allowed. The joint analysis process is applied between all relevant scores and the audio recording. The automatic description is summarized and updated in each iteration.[34]

6.13 Automatic Description of the CompMusic-Makam Corpus

By applying the joint audio-score analysis described in Section 6.12, an automatic description of the CompMusic OTMM corpus is obtained. Figure 6.23 show the statistics of the joint analysis. The automatic description covers around one third of the audio recordings and music scores in the corpus. Approximately $18,000$ sections and $750,000$ notes are in the audio recordings are linked, which correspond to more than 85 hours of time-aligned audio data. The score-informed features override the relevant audio features computed in Chapter 5 in our music discovery web applications (Section 7.1.2). The time-aligned data (e.g. the notes, measures and sections in the score; the pitch contours, pitch distribution of each note and sections in the audio recording) are also displayed synchronous to the audio playback.

6.14 Conclusion

In this Chapter, joint analysis of audio recordings and music scores is described. The analysis is based on an audio-score alignment scheme proposed for OTMM. The approach is able to handle the structural differences between the audio and symbolic data. It is robust to many performance aspects such as ahenks, tempo vari-

[33] https://github.com/sertansenturk/tomato/blob/v0.9.1/tomato/joint/jointanalyzer.py

[34] https://github.com/MTG/pycompmusic/blob/f28ad58033e5387efc7e96612fbde5333adb27ca/compmusic/extractors/makam/jointanalysis.py#L89

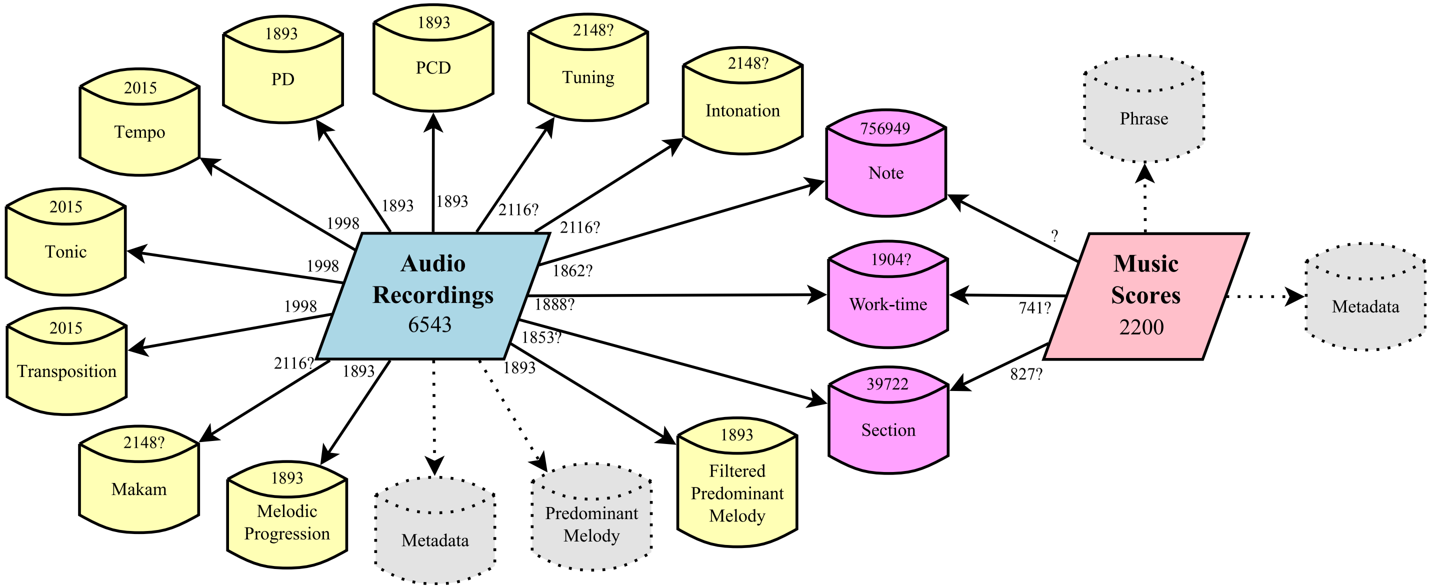


Figure 6.23: An overview of the joint description of the CompMusic OTMM corpus. The numbers in the boxes and the numbers next to the arrows indicate the total number of the instances of the relevant entity and the number audio recordings/scores for which the relevant entity is extracted. The score-informed features are shown in yellow and linked entities are shown in pink.

ations, tuning and intonation deviations, non-notated embellishments and heterophony.

The fundamental step in each task in joint audio and score analysis is partial audio score alignment (Section 6.3). This step, termed as *fragment linking*, attempts to find the time-interval(s), where a fragment picked from the music score, is performed. Fragment linking is used in joint tonic identification (Section 6.4) and tempo estimation (Section 6.5). The score-informed tonic identification method has achieved over 99% accuracy on two datasets in a wide range of parameter combinations, effectively solving the problem when the music score is available. Then, the method is extended to composition identification (Section 6.6). The proposed method is highly successful as it obtained more than 0.90 MAP for the majority of the parameter combinations with the best performing combination providing around 0.95 MAP.

To handle the structural differences between the music score and the audio recording, the music score is divided into musically relevant sections and candidate time-intervals for each section are estimated using fragment linking. The candidates are aggregated, and the best possible section sequence is inferred using graph operations (Section 6.7). This method achieves a 0.93 F_1-score in linking the sections. The results obtained from the experiments on tonic identification, composition identification and section linking also show that: **1)** predominant melody may be a better and musically more interpretable feature than HPCP in the analysis of OTMM recordings, **2)** The Hough transform may be a simpler and cheaper alternative to than DTW and HMM for audio-score alignment, when note-level precision is not needed.

The note-level alignment is obtained by applying SDTW between the predominant melody of each linked section in the audio recording and the synthetic melody computed from the relevant section in the music score (Section 6.8). Due to lack of annotations during the time of development, the note-level alignment methodology is rather simplistic compared to the rest of the proposed methodology. We have recently created the CompMusic OTMM partial audio-score alignment (Section 3.2.7) and the audio-score alignment (Section 3.2.8) datasets. Having gathered sufficient amount of data, I would like to investigate partial and complete alignment approaches based on the Hough transform, DTW and Bayesian networks (Başaran, Cemgil, & Anarım, 2015; Holzapfel et al., 2015), fingerprinting (Arzt et al., 2014) and neural networks (Raffel, 2016) applied to various computational tasks covered in this Chapter.

As a result of audio-score alignment, the sections and notes within these two different musical representations are linked with each other. The linked data may be further used to implicitly infer additional information related to lyrics and rhythm (Section 6.9) and improve existing features such as predominant melody and melodic progression (Section 6.10), and compute more accurate and informative features to describe tuning and intonation information (Section 6.11).

The CompMusic OTMM corpus is enhanced by applying the joint audio and score analysis methodologies described in this Chapter (Section 6.12). The resultant automatic description brings more reliable information compared to the automatic description obtained throughout audio analysis described in Chapter 5. The automatic description is used to complement our web application for the discovery of OTMM. The application will be explained in detail in Section 7.1.2.

The vast amount of accurate, linked and time-aligned data paves up new research topics to investigate in MIR, computational musicology and music education. In short-term, I would like to focus on corpus-based studies to describe and discover the musical characteristics of OTMM. One interesting direction could be reproducing the tuning analysis applied in (Bozkurt et al., 2009) on the whole CompMusic OTMM corpus, and extending the findings to automatically describe intonation by using the note distributions and pitch contours obtained in Section 6.11. I would also like to extend the observations of Holzapfel (2015b) on the relations between surface rhythm and usuls by analyzing the **SymbTr** collection, with a performance-driven analysis on the same dataset using audio-score alignment. Finally, I would like to investigate the musical expressivity using score-informed methods similar to the methods applied by Abesser et al. (2016) on jazz music.

Apart from OTMM, several methodologies presented in this Chapter have been applied to other melody-dominant music traditions, namely Carnatic music (Section B.1) and Cretan music (Section B.8). The results obtained from the analysis of these three music traditions show the importance of culture-specific and knowledge-based approaches in music information processing.

Following open research best practices, most of the code (in tomato), and all datasets and experimental results presented in this Chapter are openly available (Chapter C). I hope that the availability of these resources would enable future development of computational analysis methods applied to OTMM and other music traditions, in general.

Chapter 7

Applications and Conclusions

This Chapter present the web application developed for the discovery of OTMM

7.1 Applications

Dunya comprises all the music corpora and related software tools that have been developed as part of the CompMusic project. "Dunya" means the *world* in many languages such as Arabic (which is the language of origin), Turkish, Hindi and several other languages in the Indian subcontinent. The languages constitute the *de facto* languages of Arab-Andalusian, Carnatic, Hindustani and Ottoman-Turkish makam musics, i.e. all music traditions studied under the CompMusic project except Beijing Opera. While the word is generally used to imply our planet in these languages, it is also used metaphorically to refer to a realm; in our case the *realm* of music.[1]

7.1.1 Dunya-Makam

In the context of Dunya, Dunya-makam encompasses the CompMusic OTMM corpus, the culture-aware software tools (e.g. `toma-`

[1] Dunya was suggested by Mohamed Sordo, while he and I were brainstorming together to find a inclusive name for the "outputs" of the CompMusic project.

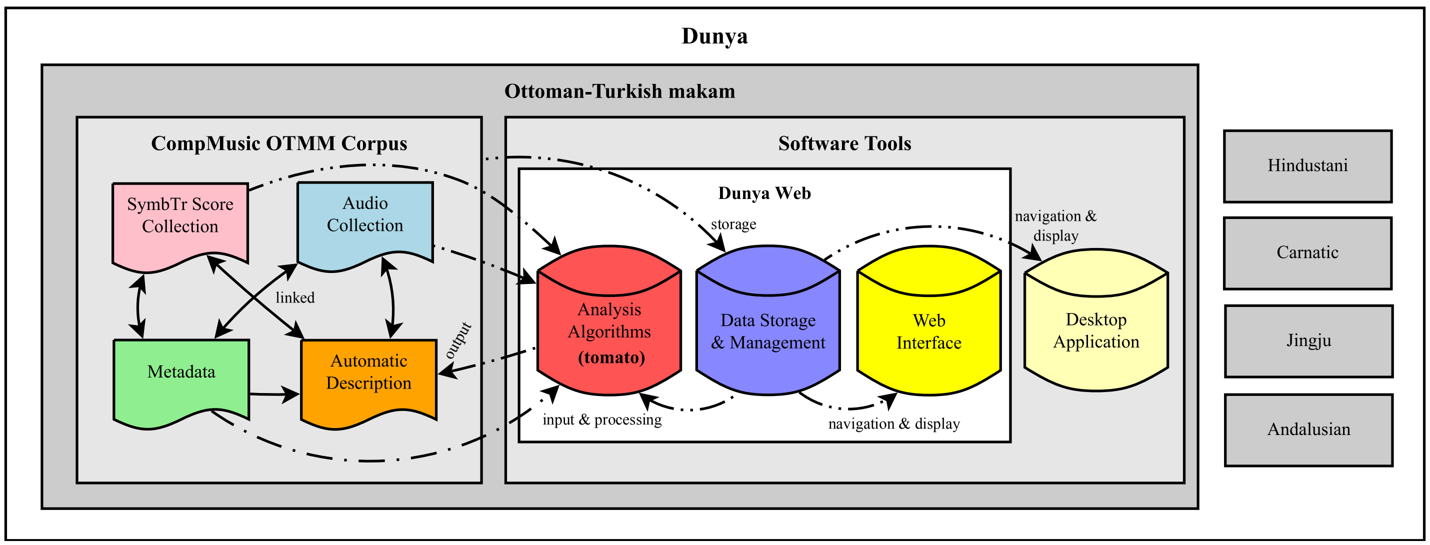


Figure 7.1: A block diagram of Dunya, focusing on the OTMM part.

to), as well as the automatic description of the corpus obtained using these tools. These resources are part of Dunya-web, a web-based application designed to store the data, the software tools and the research output, and also provide a framework to process and manage the content.

Unless otherwise indicated, all content except copyrighted material (e.g. commercial recordings) and all code in Dunya-makam and Dunya-web is licensed under CC BY-NC 3.0 (Spain) and GNU Affero General Public License Version 3 (AGPLv3), respectively.

7.1.2 Dunya Web

We have created a web application called Dunya-web to showcase our technologies developed within the CompMusic project.[2] The application stores the data, executes the algorithms (e.g. the implementations of the methodologies described between Chapters 4-6) and displays the resulting automatic analysis.[3] Dunya-web has a separate organization for each music culture studied within the CompMusic project. In this Section, we exclusively focus on the makam part of Dunya-web.

Dunya-web stores the all the audio recordings and music scores in the CompMusic OTMM corpus. The metadata is stored in MusicBrainz.[4] Apart from allowing researchers to access and maintain the resources, the website currently showcases our music discovery prototype developed for OTMM. The users can navigate the audio collection by searching or filtering by recordings, compositions, artists, makams, forms and/or usuls. When an audio recording is selected, the users are directed to the page of the recording, where they can examine the metadata, play the recordings and examine the automatic description synchronous to the audio playback (Section 7.1.2). If the **SymbTr** score of a composition performed in

[2] http://dunya.compmusic.upf.edu/makam

[3] The infrastructure has been built mainly by Alastair Porter, Andrés Ferraro and Mohamed Sordo. I have been responsible for developing and implementing the analysis methodologies described in the Chapters 4-6, integration of the algorithms to the web application and testing.

[4] Please visit http://compmusic.upf.edu/node/280 to access the data, its metadata, the code of the web application and the automatic description methodologies, the extracted features and the analysis results.

the recording is available, the score and the alignment results are displayed synchronous to the audio playback.

The navigation of the website is open to public. The metadata and the available automatic description of each recording may be freely downloaded from the relevant recording page. Due to copyright issues, playback of most recordings is only allowed to registered researchers. Nevertheless, 273 recordings in the CompMusic OTMM audio collection may be listened publicly, which we are given the streaming rights of, courtesy of (listed in the alphabetical order) Krikor Music (Ara Dinkjian), Özer Özel, Robert Garfias and Traditional Crossroads (Harold Hagopian).

Data Storage and Algorithms

Dunya-web (Porter, Sordo, & Serra, 2013b; Porter & Serra, 2014) is an application that is developed with Django framework.[5] The audio recordings, music scores and relevant metadata are stored in a PostgreSQL database.[6] It's possible to manage information about the stored data and submit analysis tasks on the data from the administration panel. The output of each analysis can be used as an input of another analysis module and/or be displayed on the interface (Section 7.1.2). The data can be accessed from the Dunya REST API. We have also developed a Python wrapper, called pycompmusic, around the Application Programming Interface (API).[7]

To render each score element synchronously in the interface (Section 7.1.2), we first convert the score in text format to MusicXML and then to SVG (Section 4.4.2). We use LilyPond for MusicXML to SVG conversion, which allows us to record a mapping of each element between these different formats. This way, each object in the SVG file can be referenced by the note that it represents in the score (Section 7.1.2).

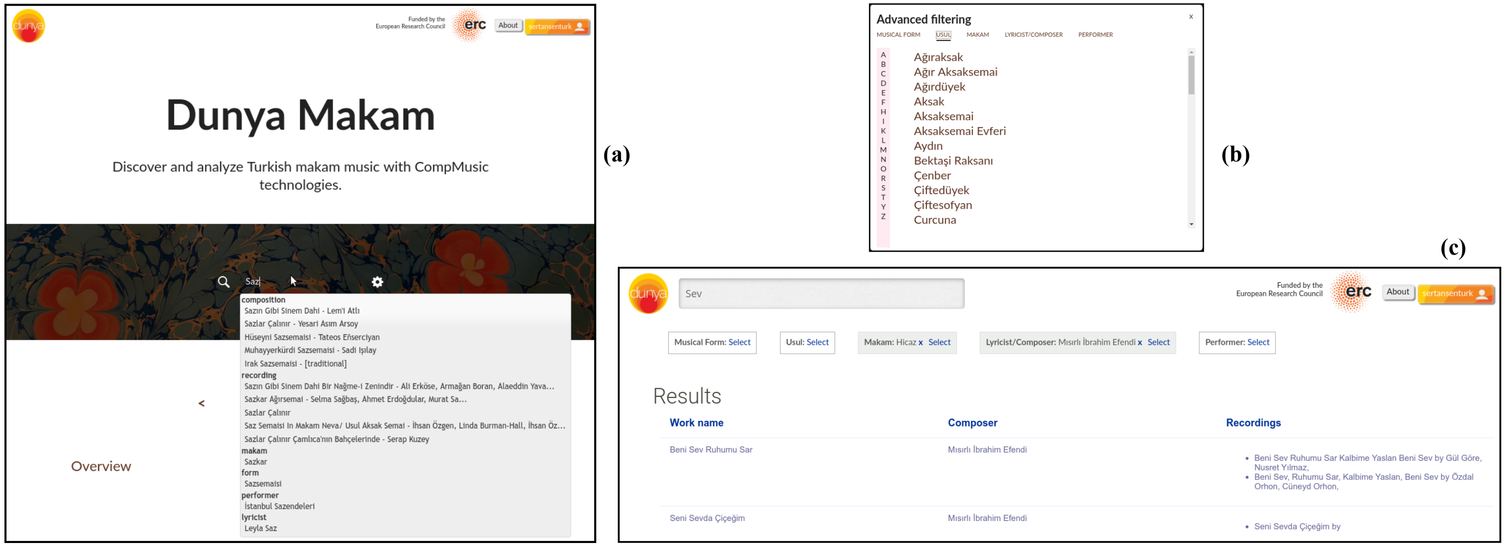
Interface

The interface interaction is developed with Javascript. In the front page, the user can search the works, makams, forms, usuls, com-

[5] https://www.djangoproject.com/

[6] https://www.postgresql.org/

[7] https://github.com/MTG/pycompmusic



Dunya Makam

Discover and analyze Turkish makam music with CompMusic technologies.

Saz

composition
Sazın Gibi Sinem Dahi - Lem'i Atlı
Sazlar Çalınır - Yesari Asım Arsoy
Hüseyni Sazsemaisi - Tateos Enserciyan
Muhayyerkürdi Sazsemaisi - Sadi Işılay
Irak Sazsemaisi - [traditional]
recording
Sazın Gibi Sinem Dahi Bir Nağme-i Zenindir - Ali Erköse, Armağan Boran, Alaeddin Yava...
Sazkar Ağırsemai - Selma Sağbaş, Ahmet Erdoğdular, Murat Sa...
Sazlar Çalınır
Saz Semaisi In Makam Neva/ Usul Aksak Semai - İhsan Özgen, Linda Burman-Hall, İhsan Öz...
Sazlar Çalınır Çamlıca'nın Bahçelerinde - Serap Kuzey
makam
Sazkar
form
Sazsemaisi
performer
İstanbul Sazendeleri
lyricist
Leyla Saz

Advanced filtering

Ağıraksak
Ağır Aksaksemai
Ağırdüyek
Aksak
Aksaksemai
Aksaksemai Evferi
Aydın
Bektaşi Raksanı
Çenber
Çifteduyek
Çiftesofyan
Curcuna

Sev

Musical Form: Select | Usul: Select | Makam: Hicaz x Select | Lyricist/Composer: Mısırlı İbrahim Efendi x Select | Performer: Select

Results

Work name	Composer	Recordings
Beni Sev Ruhumu Sar	Mısırlı İbrahim Efendi	• Beni Sev Ruhumu Sar Kalbime Yaslan Beni Sev by Gül Göre, Nusret Yılmaz, • Beni Sev, Ruhumu Sar, Kalbime Yaslan, Beni Sev by Özdal Orhon, Cüneyd Orhon,
Seni Sevda Çiçeğim	Mısırlı İbrahim Efendi	• Seni Sevda Çiçeğim by

Figure 7.2: The navigation in the Dunya-makam website. **a)** The main page with search results of query *Saz* showed instantly, **b)** The advanced filtering pop-up, accessed by clicking the cog-shaped button next to the search bar, and **c)** The search page, showing the composition results for the query “Sev” filtered by the Hicaz makam and *Mısırlı İbrahim Efendi* (`http://musicbrainz.org/artist/d8ab1cd7-262c-444f-8e6b-ee226022a316`) as the composer/lyricist.

posers, lyricists and performers by typing in the search bar, navigate to the showcased recording pages and also access to the static *Project Overview, Statistics* and *Results* pages. The results of a searched query are displayed simultaneously by Ajax[8] (Figure 7.2a). The user can also navigate by filtering the works according to these attributes without supplying a query from by selecting the attribute value from “Advanced filtering,” which can be accessed by clicking the cog-shaped button next to the search bar (Figure 7.2b). Afterwards, the user is directed to the search results (Figure 7.2c). In this page, the user can modify the query and also add, remove or modify the filters. The search is organized with respect to the works. The user can select one of the audio recordings relevant to the work. Once the user selects a recording, the recording page is opened (Figure 7.3). The page consists of four different parts:

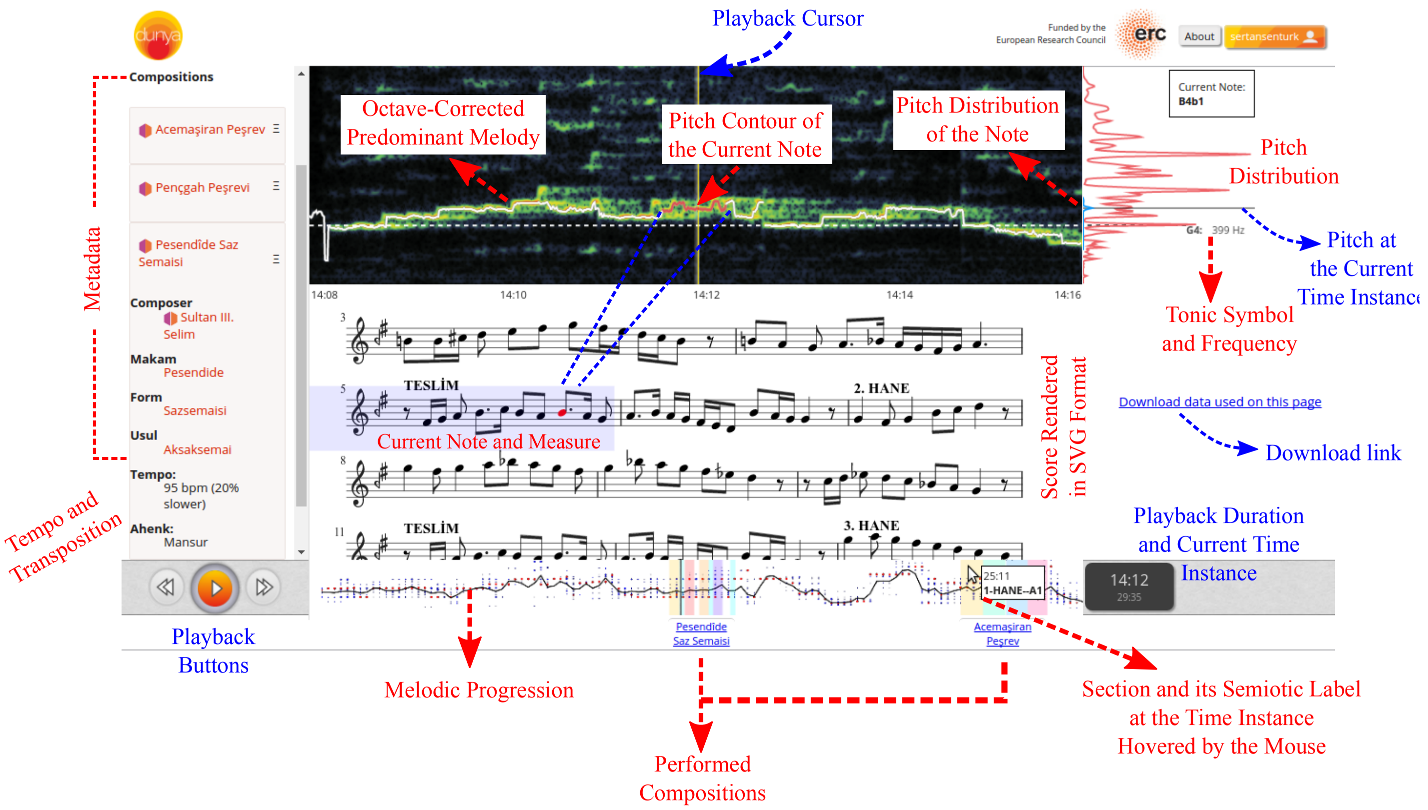
1. **Left Panel:** The metadata about the composition and the recording. Score informed tempo (Section 6.5) and transposition (Section 5.8) for each composition is also included inside the composition tabs in this panel.

2. **Top Panel:** The audio features. These include a spectrogram, the octave-corrected predominant melody (Section 6.10). The tonic frequency (Section 6.4) is drawn with a dashed line. The chunk of the predominant melody on the time-interval of the current aligned note is highlighted in red.

3. **Centre Panel:** The music score. The SVG elements corresponding to the current note and the measure are highlighted. We obtain the SVG element related to the aligned note from the mappings mentioned in Section 4.4.2. We also find the current measure by searching the two closest measure line elements, which encloses the position of the note element in the SVG file.

4. **Bottom Panel:** The playback buttons, audio timeline, playback time instance and audio duration. The melodic progression (Section 5.10) extracted from the octave-corrected predominant melody is drawn in the background of the timeline. The timeline can be clicked to jump the playback to

[8] http://api.jquery.com/jquery.ajax/



Compositions

Acemaşiran Peşrev

Pençgah Peşrevi

Pesendide Saz Semaisi

Composer: Sultan III. Selim

Makam: Pesendide

Form: Sazsemaisi

Usul: Aksaksemai

Tempo: 95 bpm (20% slower)

Ahenk: Mansur

Current Note: B4b1

dunya
Funded by the European Research Council
erc
About
sertansenturk
Playback Cursor
Octave-Corrected Predominant Melody
Pitch Contour of the Current Note
Pitch Distribution of the Note
Pitch Distribution
G4: 399 Hz
Pitch at the Current Time Instance
Tonic Symbol and Frequency
14:08
14:10
14:12
14:14
14:16
TESLİM
2. HANE
3. HANE
Current Note and Measure
Score Rendered in SVG Format
Download data used on this page
Download link
Playback Duration and Current Time Instance
14:12
29:35
25:11
1-HANE--A1
Pesendide Saz Semaisi
Acemaşiran Peşrev
Metadata
Tempo and Transposition
Playback Buttons
Melodic Progression
Performed Compositions
Section and its Semiotic Label at the Time Instance Hovered by the Mouse

Figure 7.3: The recording page of “Pençgah Solo” by Niyazi Sayın (`http://dunya.compmusic.upf.edu/makam/recording/37dd6a6a-4c19-4a86-886a-882840d59518`).

the desired time instance. If there are multiple compositions performed in the audio recording, the time intervals are indicated in the bottom of the timeline. The coloured regions mark the time-intervals of the performed sections. When the user hovers over the panel, the semiotic label of the section (Section 4.3.2) at this time is displayed.

Summary

Dunya-web is a web application, which can be used for the analysis and discovery of collections of audio recordings and music scores. So far, we have analysed 2200 music scores (Section 4.5), over 6700 audio recordings (Section 5.12) and over 1500 audio-score pairs (Section 6.13) of OTMM. We plan to incorporate score-informed rhythm analysis (Section 6.9) in the future. Currently, the audio-lyrics alignment research (Dzhambazov et al., 2014; Dzhambazov & Serra, 2015; Dzhambazov et al., 2016), is showcased in Dunya-web separately.[9] We aim to further develop these technologies to support tasks in music discovery, musicological research and music education.

7.1.3 Dunya Desktop

While the music discovery interface in Dunya-web is highly useful for navigating the corpus and examining the automatic description obtained for audio recordings, it is not designed to be customizable for different use cases. For example, the interaction is centered around individual audio recordings and it is not currently possible to access the music scores and their descriptions directly from the visual interface. Moreover, the interface does not provide any means of comparison, e.g. studying the performances of the same composition or the vocal characteristics of two singers.

Atlı (2016) has developed a visual interface to display the music scores and audio recordings of OTMM aiming at assisting music education. The visual interface is designed in Python 2.7 using Qt4. The application uses the audio analysis and joint audio-score analysis methods described in this thesis such as predominant melody

[9] e.g. http://dunya.compmusic.upf.edu/makam/lyric-align/727cff89-392f-4d15-926d-63b2697d7f3f

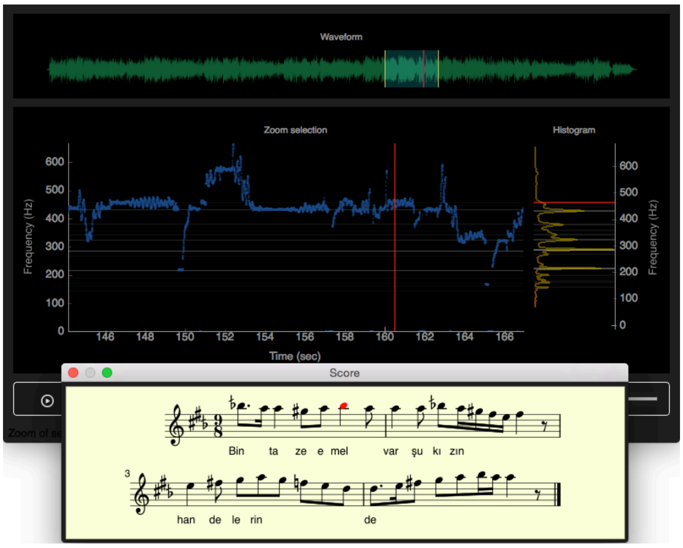


Figure 7.4: A screenshot of the interface developed by Atlı (2016) (courtesy of Hasan Sercan Atlı).

extraction (`ATL-MEL`$_f$), tuning analysis (Section 5.9), section linking (Section 6.7) and note-level alignment (Section 6.8). A screenshot of the interface is shown in Figure 7.4.

Hasan Sercan Atlı continues to develop the interface (which is called as Dunya-desktop)[10] under the CompMusic project. The aim is to address the shortcomings of the music discovery interface in Dunya-web in terms of customizability. The code is provided with detailed documentation, and it is written with modularity and extendibility of the existing modules in mind. The easily customizable interface will allow researchers to navigate through a corpus in a manner tailored to the needs of the studied problem, select the relevant material from the corpus and label/organize the content appropriately with annotations and automatic description.

7.2 Conclusions

The CompMusic OTMM corpus stands out as the largest and most representative resource of OTMM that can be used for computational research. The corpus includes 2200 music scores, more than

[10] https://github.com/MTG/dunya-desktop

6500 audio recordings, and accompanying metadata. The data has been collected, annotated and curated with the help of music experts. The potential of the research corpus is shown by using several criteria such as *completeness, coverage* and *quality*. In addition, several test datasets have been created from the corpus to develop and evaluate the specific methodologies proposed for different computational tasks addressed in the thesis.

The main contribution of the thesis is the audio-score alignment, which is designed for culture-specific properties of OTMM. The approach is able to handle the structural differences between the audio and symbolic data. It is robust to many performance aspects such as tonic transpositions, tempo variations, tuning and intonation deviations, non-notated embellishments and heterophony. The alignment method achieves a 0.93 F_1-score in linking the sections in the music score with the respective time-intervals in the audio recording. The joint analysis method not only links the audio and the symbolic data, but it also simplifies and improves the audio feature extraction steps, that would require sophisticated audio processing approaches. For example the score-informed tonic identification method achieved over 99% accuracy, effectively solving the problem when the music score is available. Likewise, the composition identification method achieved a mean average precision around 0.95.

The analysis methodologies presented in the thesis are implemented within a comprehensive and easy-to-use toolbox in Python. The algorithms are applied to the CompMusic OTMM corpus and an automatic description of the corpus is obtained. The results are integrated into Dunya-web, a web application aimed at culture-aware music discovery. Several of the methodologies developed within the thesis are also applied to other musical cultures. Following open research best practices, all the created data, software tools and analysis results are openly available. The methodologies, the tools and the corpus itself provide vast opportunities for future research in many fields such as music information retrieval, computational musicology and music education.

Appendix **A**

Preliminary Section Linking Methodology

In this Appendix, the preliminary section linking methodology proposed in (Şentürk et al., 2012) is decribed. The method uses a machine readable version of the score of a composition (selected from the **SymbTr** collection) and an audio recording consisting of a performance of the same composition as the inputs. The method also utilizes complementary metadata about these information sources and related concepts from makam music theory (Section A.1). From the audio recording, the predominant melody is extracted. The predominant melody is also used to calculate a pitch histogram in order to identify the tuning and the note intervals (Section A.1.1). From the score information, the note symbols, the sections and the makam are read, and a synthetic predominant melody is generated (Section A.1.2). In order to estimate the candidate locations of the sections in the audio, the method compares these relevant pitch representations (Section A.2). In the final step, the candidates are hierarchically checked to link the sections of the score to the corresponding parts in the audio (Section A.3). The block diagram of the methodology is given in Figure A.1.

This method has been tested on a small dataset consisting 44 audio recordings of 11 music scores of peşrev and sazsemaisi forms (Section A.4) and shown to provide promising results (Section A.5). This method has been improved and simplified in (Şentürk, Holzapfel, & Serra, 2014), which was described in Section 6.7.

A.1 Feature Extraction

To link the identified score sections with their performances, machine-readable scores and audio recordings are used. These information sources are already associated with each other through complementary metadata available, so that there is no need to apply composition identification (Section 6.6) prior to section linking. The scores are encoded as **SymbTr** files (Karaosmanoğlu, 2012), a Humdrum-like machine readable format. The starting and ending of the sections are explicitly marked in the scores. Some theoretical knowledge, namely the letter symbols of the notes, the letter symbol of the karar note of the makam of the piece and melodic intervals are used to process the audio recordings and the symbolic scores, which will be explained in Section A.1.1 and Section A.1.2.

A.1.1 Predominant Melody Extraction and Tuning Analysis on the Audio Recordings

To obtain the predominant melody from the audio recordings, the predominant melody extraction procedure described in (Şentürk et al., 2012), which is adapted from (Bozkurt, 2008) ($\texttt{BOZ-YIN}_f$) is used (explained in Section 5.2.1). The tonic is identified using the implementation of the distribution matching method (Gedik & Bozkurt, 2010) (explained in Section 5.7.2) in the Makam Toolbox. Next, the predominant melody is normalized with respect to the identified tonic. The pitch values in the time-intervals without any estimation (e.g. noise, silence) are assigned a non-sensical numerical value. In parallel, a histogram is computed from the predominant melody (Section 5.5) with a bin width of $1/3$ Hc. Next, tuning analysis (Section 5.9) is applied to the histogram to obtain the performed intervals for each note symbol.

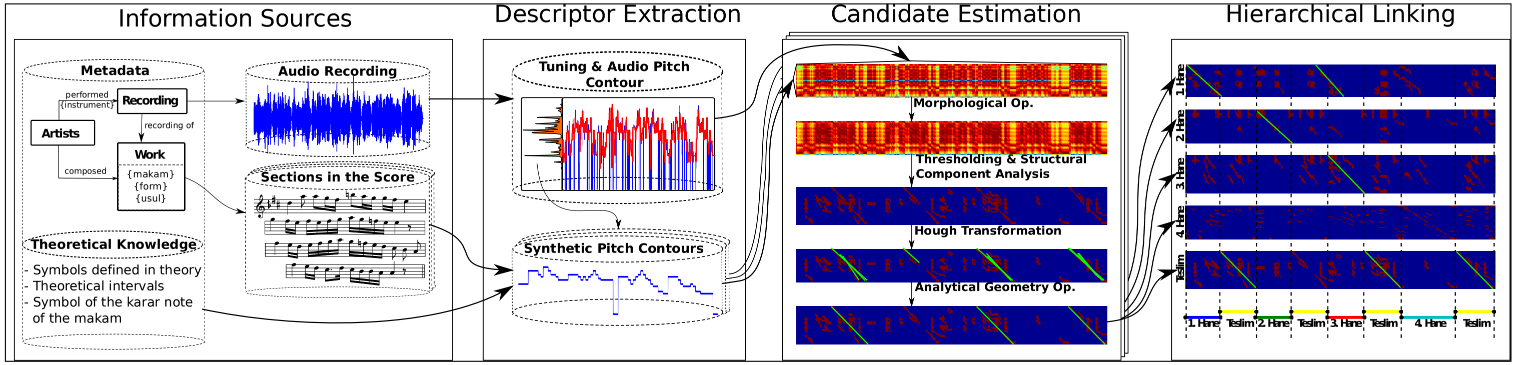


Figure A.1: Block diagram of the section linking methodology between a score of a piece and an audio recording of the same piece.

Table A.1: Structural element defined for the dilation operation.

1	1	1	.	.	.	.	.	.
1	1	1	.	.	.	.	.	.
1	1	1	1	.	.	.	.	.
.	.	1	1	.	.	.	.	.
.	.	.	.	1	.	.	.	.
.	.	.	.	.	1	1	.	.
.	.	.	.	.	1	1	1	1
.	.	.	.	.	.	1	1	1
.	.	.	.	.	.	1	1	1

A.1.2 Synthetic Pitch Contour Generation on the Music Scores

From the score, the makam of the piece, the starting event numbers of the sections, the note names and their durations are read. If the repetitive section have different endings, only the note sequence of the first instance is considered. The symbolic format is first mapped to theoretical pitches with respect to the theoretical information given (as described in Section 4.2.2), such that the karar note is assigned to 0 Hc and all note symbols are converted to their respective theoretical scale degree values (i.e. the symbol $B4\flat2$ is converted to 7 Hc, where the karar note of a piece is $A4 = 0$ Hc). Then each value obtained from the theoretical intervals is interchanged with the scale degrees in the performance obtained through tuning analysis (Section A.1.1). The rests in the score are assigned the same nonsensical value, which was noted in audio predominant melody extraction (Section A.1.1). Then, the note and time sequences are divided into sections by using the event number of the start of each section. Finally a synthetic predominant melody of each section is generated (Section 4.2.2) from the durations and the Hc values of the note sequences (obtained from tuning analysis) in the segments with a sampling period of 100ms to match the hop size of the downsampled audio predominant melody.

Table A.2: Structural element defined for the erosion and opening operations.

1	1	.	.	.
1	1	.	.	.
.	.	1	.	.
.	.	.	1	1
.	.	.	1	1

A.2 Candidate Estimation

After feature extraction, a distance matrix is computed between the audio predominant melody and the synthetic melody of each score section using Equation 6.1. The distance matrices are then normalized so that the values are between 0 and 1.

In the normalized distance matrices, long, diagonal “valleys” (i.e. adjacent distance values close to zero) are observed, which identify the time-intervals in the audio recordings, where the selected section in the score might be performed. Prior to detection, First these blobs are emphasized by utilizing a number of structural morphological operations (Serra, 1983; Ballard, 1981). To properly apply morphological operations, the values in the distance matrices are subtracted from 1 such that values close to 0 in the distance matrix are mapped close to 1, and vice versa. Then, the matrix is dilated using a binary diagonal beam shown in Table A.1 as the structural element. Afterwards, the distance matrix is eroded twice using a similar but smaller beam as shown in Table A.2. Later, the distance matrix is opened with the same structuring element used in the erosions Table A.2 to remove noises. Next, the distance matrices are converted into binary images by applying thresholding, such that all values higher than 0.96 are given the value one and all other values are assigned to zero. Structural component analysis is done on the binary image to find the blobs. All blobs that are not in the desired diagonal orientation (i.e. lying between 0 and -90 degrees) are removed. From the remaining blobs only the biggest 20% are picked. As a last step in pre-processing the distance matrix, the image is dilated by a 3×3 square structuring element to slightly widen the diagonals.

After pre-processing the distance matrices, Hough transform

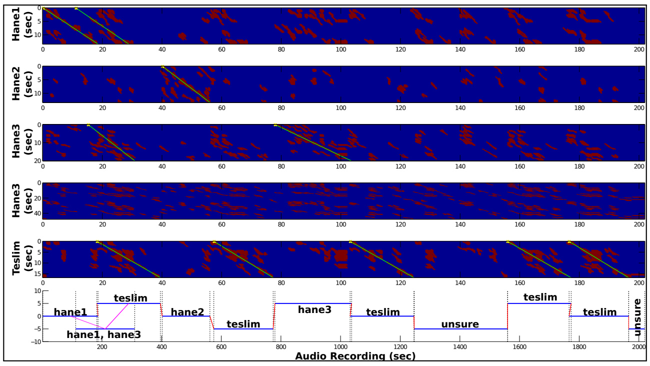


Figure A.2: Section candidates shown on top of the processed distance matrices, estimated for an audio recording of Muhayyer Saz Semâi (recording #29 in Table A.3) and groups connected prior to sequential linking. Horizontal blue lines show the group borders, red lines indicate connections of preceding and following groups and pink links mark overlapping regions.

(Ballard, 1981) is applied on each distance matrix to detect the prominent lines. The peaks between -25 and -65 degrees are detected in the transformation matrix, and the peaks which have accumulated a value higher than 0.3 are picked. The detected peaks are then used to extract line segments: in this process only the lines which are longer than 150 pixels are selected. Since the diagonals are actually blobs, there are a number of lines in the same region with small variances in locations and angles: all of these lines are removed except the longest one. Moreover, some prominent diagonals might have discontinuities resulting in more than one line segment on different parts of a diagonal. These lines are connected with each other provided their combined projection to the score (i.e. the range in the corresponding y-axis) covers more than 60% of the score. Finally, all line segments covering more than 70% of the score are extrapolated to the edges and all other lines are removed. By combining the parallel results, candidate locations for all sections are obtained.

A.3 Sequential Linking

Through inspecting the candidates obtained from the estimations of each section, most of the sections may be linked with their corresponding regions in the audio recording. Nevertheless, there might be some erroneous candidates in several locations apart from the true location. Since the candidate estimations for each section are temporally independent from each other, such erroneous links might overlap or enclose other candidates, and produce conceptually problematic outcomes. Moreover, there might also be some unsure regions where no candidate was estimated.

Nevertheless, since the sequence of the sections in the score is known, an additional step making use of the sequence of the sections given in the composition might be introduced. This step would be hierarchically able to eliminate any erroneous candidates and guess unsure regions, and therefore increase the overall accuracy of the method.

First, the candidates are gathered such that when the borders of a candidate is inside the borders of another (i.e. one candidate is enclosing another), they are grouped together. Since there is always a chance for the shorter candidate to be exceeding a border of the longer candidate by a very small duration, an expansion outside the border of the longer candidate by less than 10% of the duration of the longest candidate is tolerated. Next, regions, where candidate estimation did not predict any candidates, are labeled as "unsure." Afterwards, these groups are connected together so that any preceding, following and overlapping groups may be traversed (Figure A.2).

After the enclosing groups are formed, linking is commenced iteratively. First, any non-overlapping groups having a single candidate are temporarily linked. Next, each Hane candidate is checked whether its location is impossible with respect to already linked candidates. For example, if a 2nd Hane is linked and there are other 2nd Hane candidates occurring later in the audio recording, which are not directly connected to the link (i.e. a sequence of $\{2^{\text{nd}}\text{ Hane}, 2^{\text{nd}}\text{ Hane}\}$ is not observed) or through an unsure region (i.e. a sequence of $\{2^{\text{nd}}\text{ Hane}, \text{unsure}, 2^{\text{nd}}\text{ Hane}\}$ is not observed), these future candidates are removed even if they are already linked. Moreover any earlier candidates which should not occur before a Hane link (i.e. 3rd Hane and 4th Hane candidates occurring before a 2nd Hane link) or should not occur after a Hane link (i.e. 1st Hane candidates occurring after a 2nd Hane link) are removed. This way, most of the false positives occurring before and after the true Hane link may be taken care of, while linking the Hane repetitions. Note that the method allows sub-performances between two sections with the same label, which are not related to the composition (e.g. taksim).

After this step, the indices of links (i.e. order of the section given in the score) are noted, where possible. Since each Hane has an unique index in the score, our starting point is to note the indices of the linked Hanes. For example, if the score is in the form [1st Hane, Teslim, 2nd Hane, . . . , 4th Hane, Teslim], the index of a 2nd Hane link will be 3. If a Teslim or a Teslim repetition is found, the index will be the index of the previous neighboring Hane plus one or the index of the next neighboring Hane minus one, provided either one is known. If the indices of both the previous and the next neighboring Hane link is known, they must be consecutive (i.e. [1st Hane, Teslim(s), 2nd Hane]), or the indices for the Teslim will be left indeterminate. The indices of the links are used to estimate the unsure groups and groups with mulitple candidates, which will be explained later.

Through inspecting the enclosing groups, it was seen that if a group is overlapping with at least two other groups, the candidates inside the group are almost never true positives. All such overlapping groups are removed to increase precision in exchange with a minimal-to-zero decrease in recall.

After each step, if all the candidates of an enclosing group is removed, the group is assigned “unsure.” Moreover, if an unsure group is followed by another, both groups are merged into one. Unsure groups are also not allowed to overlap with other groups. If such a case occurs the interval overlapping with the other groups is trimmed from the unsure group.

The final confusion arises when a group does not have any candidates (unsure group) or there are at least two candidates that are both linkable. To guess an unsure group, both of the immediate neighbor groups must be already linked.[1] If the neighbors are

[1] With the exception of the first and the last groups since they are in the start

consecutive Hanes, the algorithm predicts a Teslim for the unsure group. If both of the neighbors are Teslims, the algorithm predicts a Hane in between, provided that at least one of the composition index of the (Teslim) neighbors are previously noted. If both indices are known, they must be even consecutive[2] so that there can only be a single Hane nominee. If these conditions are not met and only one of the neighbors is a Teslim, the algorithm predicts a Teslim repetition. Otherwise, the group is left as unsure. For groups, which multiple candidate are possible, the same operation is done. Nevertheless, a multiple-candidate group only requires a single neighbor to be linked before. Moreover, if the unlinked neighbor has more than one candidate (i.e. it is also a multi-candidate group), all candidates in this neighboring group are considered one-by-one to link the multi-candidate group.

The iterative process is finished if no border changes or linking is done in a cycle. Afterwards the gaps between each neighboring link are closed provided there is one. The first and the final links are also widened to the start and the end of the audio recording provided the are not further from the start/end more than 10% of the duration of the longest candidate. Finally, all of the remaining unsure regions are converted to links indicating regions which indicate unrelated parts in the performance with respect to the given composition.

A.4 Experiments

To test the methodology, we have gathered scores of instrumental pieces and the corresponding audio recordings (Section A.4.1). The method is applied to each audio recording, linking the sections marked in the score with the corresponding audio fragments. The links found between the audio recordings and scores are then compared with manually linked regions (Section A.5).

and end of the recording respectively. For the first and the last groups respectively, only the next and previous groups are needed to be linked before.

[2] Since both sazsemaisi and peşrev forms start with 1^{st} *hane*, Teslims always occupy even indices.

Table A.3: The dataset used in the experimentation. h_n, t and u stand for the n^{th} Hane, Teslim and unrelated region respectively. t* indicates ends of the Teslims vary in the composition.

Rec. #	Composition	Composer	Instrumentation	Dur.	Annotations	Comments
1	Acemaşiran Peşrev	Neyzen Salih Dede	Ney	4:19	h_1, h_2, h_3, h_4	Kız Ahenk
2			Ney	4:22	h_1, h_2, h_3, h_4	Kız Ahenk
3			Ney	4:22	h_1, h_2, h_3, h_4	Mansur Ahenk
4	Hicaz Saz Semâî	Muhittin Erev	Ney	4:00	h_1, t, h_2, t, h_3, t, h_4, t	Kız Ahenk
5			Ney	4:00	h_1, t, h_2, t, h_3, t, h_4, t	Mansur Ahenk
6	Hüseyni Peşrev	Kul Mehmet	Ney	5:21	h_1, h_2, h_3, h_4	Kız Ahenk
7			Ney	5:22	h_1, h_2, h_3, h_4	Mansur Ahenk
8	Hüseyni Peşrev	Lavtacı Andon	Ensemble	5:17	h_1, t*, h_2, t*, h_3, t*, h_4, t*, u	Silence in the End
9			Ensemble	5:15	h_1, t*, h_2, t*, h_3, t*, h_4, t*, u	Silence in the End
10	Hüseyni Saz Semâî	Lavtacı Andon	Ney	4:48	h_1, t, h_2, t, h_3, t, h_4, t	Kız Ahenk
11			Ney	4:48	h_1, t, h_2, t, h_3, t, h_4, t	Mansur Ahenk
12	Hüseyni Saz Semâî	Tatyos Efendi	Ensemble	3:01	h_1, t, h_2, t, h_3, h_3, t, h_4, t, t, u	Silence in the End
13			Ensemble	5:38	h_1, t, t, h_2, h_2, t, t, h_3, h_3, t, t, h_4, t, t, u	Silence in the End
14			Tanbur, Kemençe	3:21	h_1, t, t, h_2, t, h_3, h_3, t, t, h_4, t, t, u	Repetitions in Hane 4 Omitted Silence in the End
15			Ud	7:31	u, h_1, h_1, t, t, h_2, h_2, t, t, h_3, h_3, t, t, h_4, t, t, u	Speech and Taksim in the Start Taksim and Silence in the End
16	Kürdilihicazkar Peşrev	Vasilaki	Ensemble	1:10	h_1, t*	Partial Performance
17			Ensemble	1:11	h_1, t*	Partial Performance
18			Tanbur	4:05	h_1, t*, h_2, t*, h_3, t*, h_4, t*, u	Denoised Recording of Below Silence in the End
19			Tanbur	4:07	h_1, t*, h_2, t*, h_3, t*, h_4, t*, u	Noisy Recording Silence in the End
20			Ud	4:19	h_1, t*, h_2, t*, h_3, t*, h_4, t*, u	Silence in the End
21			Ensemble	5:48	h_1, t*, h_2, t*, h_3, t*, h_4, t*, u	Silence in the End
22			Ensemble	2:07	h_1, t*, h_2, t*	Partial Performance
23	Muhayyer Saz Semâî	Tanburi Cemil Bey	Ud	6:32	u, h_1, t, t, h_2, t, t, h_3, t, t, h_4, t, t, u	Silence in the Start and the End
24			Ud	4:08	h_1, t, t, h_2, t, t, h_3, t, t, h_4, t, t, u	Silence in the End
25			Ud	4:16	h_1, t, t, h_2, t, t, h_3, t, t, h_4, t, t, u	Silence in the End
26			Ensemble	5:33	h_1, t, t, h_2, t, t, h_3, t, t, h_4, t, t, u	Silence in the End
27			Ney	4:20	h_1, t, h_2, t, h_3, t, h_4, t	Kız Ahenk
28			Ney	4:20	h_1, t, h_2, t, h_3, t, h_4, t	Mansur Ahenk
29			Ensemble	3:22	h_1, t, h_2, t, h_3, t, h_4, t, t, u	Silence in the End
30	Rast Peşrev	Osman Bey	Ney	4:10	h_1, t, h_2, t, h_3, t, h_4, t	Kız Ahenk
31			Ney	4:09	h_1, t, h_2, t, h_3, t, h_4, t	Mansur Ahenk
32	Segah Saz Semâî	Yusuf Paşa	Ensemble	2:36	h_1, t*	Partial Performance
33			Violin	7:35	u, h_1, t*, h_2, t*, h_3, t*, h_4, t*, u	Silence in the Start and the End
34			Ney, Percussion	3:27	h_1, t*, h_2, t*	Percussion is Recorded Loud
35			Cello, Viola	14:03	h_1, t*, h_2, t*, h_3, t*, h_4, t*, u	Group Taksim, Suzidil Saz Semaisi and Silence in the End
36			Ney, Kanun	6:39	h_1, t*, h_2, t*, h_3, t*, h_4, t*	
37	Uşşak Saz Semâî	Salih Dede	Tanbur	6:45	h_1, t, t, h_2, t, t, h_3, t, t, h_4, h_4, t, t	
38			Tanbur, Kemençe	4:16	h_1, t, h_2, t, h_3, t, h_4, t, u	Silence in the End
39			Ud	5:53	h_1, t, t, h_2, t, t, h_3, t, t, h_4, t, t	
40			Tanbur	5:44	h_1, t, t, h_2, t, t, h_3, t, t, h_4, t, t, u	Silence in the End
41			Kemençe	5:20	h_1, t, h_2, t, h_3, t, t, h_4, u, h_4, t, t, u	Taksim in the Middle Silence in the End
42			Ney	5:56	h_1, t, h_2, t, h_3, t, h_4, t	Kız Ahenk
43			Ney	5:56	h_1, t, h_2, t, h_3, t, h_4, t	Mansur Ahenk
44			Ney	7:16	h_1, t, t, h_2, t, t, h_3, t, t, h_4, t, t	Müstahsen Ahenk

A.4.1 Data

For the experiments we have used a set of 44 audio recordings associated with 11 scores of different compositions (Table A.3). The scores and parallel audio recordings come from the **SymbTr** collection (Karaosmanoğlu, 2012) and the *CompMusic*-makam corpus, respectively.

All the scores follow the Arel-Ezgi-Uzdilek theory. In the experiments, we are using a single score per composition, which is either obtained from the **SymbTr** collection (Karaosmanoğlu, 2012). As score fragments, we use the actual sections of the pieces, a total of 53 fragments. All of the audio recordings are in *wav* format and either public-domain[3] or commercially available. The recordings encompass a wide variety of instrumentation (Table A.3) such as solo ney recordings, which are monophonic; solo stringed instruments, which involve heterophonic peculiarities; duo, trio and ensembles, which are heterophonic. The recordings also cover a substantial amount of expressive decisions such as changes in performance speed, different density of embellishments, note suspension and repetitions, melodic excerpts played in different octaves and various ahenks. Some of the recordings include some material that is not related to the scores such as taksims, applauses, introductory speeches, silences and even other pieces of music. These audio materials are not manually removed.

A.5 Results and Evaluation

To evaluate the method, we built the ground truth by manually identifying the particular fragment of the score section by labeling the time boundaries in the audio recordings. A composition-related link is deemed as true positive, if and only if it is coinciding with an annotation of the same section, and the average distance between the borders of the annotation and the link does not exceed 10% of the duration of the annotation. Links, which do not meet these constraints are treated as false positives. If a composition related annotation does not coincide with any link with the distance constraint given above, it is labeled as a false negative.

Since the system is not meant to identify what a non-related region actually is, the boundaries of the links labeled as "unrelated" do not have to coincide with the borders of an unrelated annotation. Therefore, any consecutive unrelated regions (i.e. introductory speech followed by a taksim) are combined into a single one, and evaluation is done on the links which are enclosed by a

[3] e.g. the *Instrumental Pieces Played with the Ney* collection: http://neyzen.com/ney_den_saz_eserleri.htm

Table A.4: The results per piece. t and t_N indicate the time and normalized time elapsed per experiment with semi-automatic karar recognition. *K*-, *K*+, *H*- and *H*+ indicate results obtained from fully-automatic karar recognition, semi-automatic karar recognition, candidate estimation and sequential linking respectively.

	#Sections /	t / t_N	True Positive				True Negative				False Negative				False Positive			
Rec. #	**#Unrelated**	**(sec)**	**K-H-**	**K+H-**	**K-H+**	**K+H+**	**K-H-**	**K+H-**	**K-H+**	**K+H+**	**K-H-**	**K+H-**	**K-H+**	**K+H+**	**K-H-**	**K+H-**	**K-H+**	**K+H+**
1	4	32 / 34	**0**	4	**0**	4	0	0	0	0	**4**	0	**4**	0	0	2	0	0
2	4	26 / 27	**0**	4	**0**	4	0	0	0	0	**4**	0	**4**	0	0	3	0	0
3	4	26 / 28	**0**	4	**0**	4	0	0	0	0	**4**	0	**4**	0	0	3	0	0
4	8	28 / 32	7	7	8	8	0	0	0	0	1	1	0	0	0	0	0	0
5	8	33 / 37	7	7	8	8	0	0	0	0	1	1	0	0	1	1	0	0
6	4	39 / 33	**0**	4	**0**	4	0	0	0	0	**4**	0	**4**	0	0	1	0	0
7	4	39 / 33	**0**	4	**0**	4	0	0	0	0	**4**	0	**4**	0	0	0	0	0
8	8 / 1	30 / 26	**0**	3	**0**	5	0	0	0	0	**8**	5	**8**	3	0	0	0	1
9	8 / 1	32 / 28	7	7	8	8	0	0	0	0	1	1	0	0	0	0	0	0
10	8	27 / 32	7	7	8	8	0	0	0	0	1	1	0	0	1	1	0	0
11	8	27 / 32	8	8	8	8	0	0	0	0	0	0	0	0	3	3	0	0
12	10 / 1	28 / 42	4	4	5	5	0	0	1	1	6	6	5	5	1	1	2	2
13	14 / 1	67 / 66	10	10	12	12	0	0	0	0	4	4	2	2	0	0	2	2
14	12 / 1	46 / 62	11	11	10	10	0	0	0	0	1	1	2	2	2	2	2	2
15	15 / 2	126 / 89	13	13	14	14	0	0	2	2	2	2	0	0	1	1	3	3
16	2	13 / 53	2	2	2	2	0	0	0	0	0	0	0	0	0	0	0	0
17	2	14 / 52	1	1	2	2	0	0	0	0	1	1	0	0	0	0	0	0
18	8 / 1	30 / 34	7	7	7	7	0	0	0	0	1	1	1	1	0	0	1	1
19	8 / 1	28 / 31	5	5	6	6	0	0	0	0	3	3	2	2	0	0	0	0
20	8 / 1	29 / 30	5	5	6	6	0	0	0	0	3	3	2	2	0	0	1	1
21	8 / 1	32 / 30	4	4	8	8	0	0	0	0	4	4	0	0	0	0	0	0
22	4	17 / 37	**0**	2	**0**	4	0	0	0	0	**4**	2	**4**	0	0	0	0	0
23	12 / 2	40 / 33	5	5	7	7	0	0	1	1	7	7	5	5	0	0	2	2
24	12 / 1	32 / 36	4	4	7	7	0	0	0	0	8	8	5	5	1	1	1	1
25	12 / 1	59 / 63	7	7	8	8	0	0	0	0	5	5	3	3	1	1	5	5
26	12 / 1	50 / 50	7	7	10	10	0	0	0	0	5	5	2	2	0	0	2	2
27	8	28 / 29	7	7	8	8	0	0	0	0	1	1	0	0	0	0	0	0
28	8	31 / 33	7	7	7	7	0	0	0	0	1	1	1	1	1	1	2	2
29	9 / 1	40 / 54	8	8	9	9	0	0	0	0	1	1	0	0	2	2	1	1
30	8	33 / 36	8	8	8	8	0	0	0	0	0	0	0	0	3	3	0	0
31	8	36 / 39	8	8	8	8	0	0	0	0	0	0	0	0	5	5	0	0
32	2	15 / 44	2	2	2	2	0	0	0	0	0	0	0	0	1	1	0	0
33	8 / 2	45 / 32	7	7	8	8	0	0	1	1	1	1	0	0	4	4	0	0
34	4	23 / 31	4	4	4	4	0	0	0	0	0	0	0	0	0	0	0	0
35	8 / 1	101 / 33	5	5	5	5	0	0	0	0	3	3	2	2	2	2	4	4
36	8	44 / 36	8	8	8	8	0	0	0	0	0	0	0	0	3	3	1	1
37	13	76 / 61	12	12	12	12	0	0	0	0	1	1	0	0	0	0	2	2
38	8 / 1	32 / 34	7	7	7	7	0	0	0	0	1	1	1	1	2	2	1	1
39	12	61 / 57	11	11	12	12	0	0	0	0	1	1	0	0	0	0	1	1
40	12 / 1	93 / 90	10	10	10	10	0	0	1	1	2	2	2	2	1	1	1	1
41	11 / 2	63 / 54	9	9	10	10	0	0	0	0	2	2	1	1	0	0	2	2
42	8	39 / 37	8	8	8	8	0	0	0	0	0	0	0	0	0	0	1	1
43	8	41 / 38	8	8	8	8	0	0	0	0	0	0	0	0	2	2	1	1
44	12	69 / 44	12	12	12	12	0	0	0	0	0	0	0	0	3	3	1	1
Total	364 / 24	1817 / 1831 (**Av:** 41 / 42)	262	287	290	319	0	0	6	6	100	75	68	39	40	49	39	40

non-compositional region. Links enclosed by a non-compositional region are obtained by the enclosing operation explained in Section A.3. All links labeled as “unrelated” enclosed by a non-compositional annotation are labeled as true negative. All other enclosed links are treated as false positives. Any unguessed parts in these annotations are neither awarded or penalized.

We have computed accuracy, specificity, recall, precision, F_1-score and F_3-score from the true positives, true negatives, false positives and false negatives. These results are reported for both candidate estimation and sequential linking. The automatic karar recognition obtained via Makam Toolbox has failed in 7 pieces (recordings #1, #2, #3, #6, #7, #8 and #22, indicated as bold in Table A.4),

Table A.5: The results of the section linking experiment including all audio recordings. *K-*, *K+*, *H-* and *H+* indicate results obtained from fully-automatic karar recognition, semi-automatic karar recognition, candidate estimation and sequential linking respectively.

	K-H-	**K+H-**	**K-H+**	**K+H+**
Accuracy	65.17%	69.83%	73.45%	80.45%
Specificity	0%	0%	13.33%	13.04%
Recall	72.38%	79.28%	81.01%	89.11%
Precision	86.75%	85.42%	88.15%	88.86%
F_1 score	78.92%	82.23%	84.43%	88.98%
F_3 score	73.60%	79.86%	81.67%	89.08%

Table A.6: The results obtained from the candidate estimation with semi-automatic karar detection. The results are grouped per instrumentation. *#Rec., #Sec., #Un., tp, fn, fp, Accur., Precis., F_1, F_3* stand for number of recordings, number of sections, number of unrelated regions, number of true positives, number of false negatives, number of false positives, accuracy, precision, F_1-score and F_3-score respectively.

	#Rec.	**#Sec.**	**#Un.**	**tp**	**fn**	**fp**	**Accur.**	**Recall**	**Precis.**	**F_1**	**F_3**
Solo Ney	17	116	0	111	5	28	77.08%	95.69%	79.86%	87.06%	93.83%
Solo Stringed	12	131	14	95	36	8	68.35%	72.52%	92.23%	81.20%	74.10%
Duo / Trio	4	36	3	31	5	9	68.89%	86.11%	77.50%	81.58%	85.16%
Ensemble	11	79	7	50	29	4	60.24%	63.29%	92.59%	75.19%	65.36%
All	44	362	24	287	75	49	69.83%	79.28%	85.42%	82.23%	79.86%

which are corrected via the graphical interface of the Makam Toolbox. The true positive, true negative, false positive, false negative scores calculated per experiment is given in Table A.4. The global accuracy, specificity, recall, precision, F_1 score and F_3 score obtained from the candidate estimation and sequential linking with automatic and semi-automatic karar recognition are given in Table A.5.

In order to assess the effectiveness of predominant melodies proposed, it is necessary to check the results obtained from the candidate estimation with respect to the density of heterophonic and expressive elements. However, it is not straightforward to directly measure the level of heterophony and expressivity of an audio recording. On the other hand, since these elements are related to instrumentation, the results obtained from candidate estimation are grouped and compared with respect to different types instrumentation (Table A.6).

The time elapsed per experiment are also recorded. The timings are then normalized with respect to the duration of the audio recordings with the given formula:

$$t_{Ni} = \frac{t_i}{dur_i} * \frac{\sum_i^n dur_i}{n} \tag{A.1}$$

where t_i is the time elapsed during the section linking, dur_i is the duration of the i^{th}audio recording and n is the number of the recordings (Table A.5). It takes an average of 42 seconds with a standard deviation of 15 seconds to link the sections of a audio recording approximately 275 seconds long (i.e. the average duration of an audio recording in the dataset), when the implementation is run on computer with a 4 GB RAM and 2.26 GHz processor.

A.6 Discussion

The results in Table A.5 points that the methodology is successful in linking sections given in the scores with the corresponding audio recordings. The method is able to deal with a wide number of situations such as compositions without any section repetitions, various ahenks, partial performances, Hane or Teslim repetitions and recordings with unrelated parts. Table A.5 also shows that sequential linking has a clear success over candidate estimation, even when failed karar detections are not altered.

The advantage of the sequential linking is more evident, when results per piece (Table A.4) are inspected. Except the 14^{th} recording, where candidate estimation produced one erroneous link enclosing a true link and sequential linking preferred the erroneous one, sequential linking emits more true positives and less false negatives. Moreover, there is no increase in the number of false positives obtained through all experiments, thus sequential linking presents much better precision, recall and F-scores over evaluation on raw links provided by the section estimation.

The results also show that the predominant melodies computed by `BOZ-YIN`$_f$ is a representative feature for section linking applied to OTMM. Nevertheless, in Table A.6, it can be seen that as the instrumentation of a recording gets more complex, i.e. the tendency of observing heterophonic and expressive elements in an audio recording increases, the accuracy and the F_1-score decreases almost monotonically. This suggests that an improvement in the extraction of audio predominant melody is necessary. Through inspecting errors in the audio recording level, it is seen that the current bottleneck of the system is the pitch estimation. Since `YIN` is designed for monophonic sounds, lots of confusions arise in the fundamental frequency estimations due to the heterophonic nature of OTMM, especially in ensemble performances. Moreover, `YIN` is found to lose its robustness, where there are substantial usage of expressive elements such as legatos, slides and tremolos. This problem should be tackled by using multi-pitch extraction and prominent melody detection (Salamon & Gómez, 2012).

A second problem occurs when the performers substantially deviate from the score i.e. a performer suspends the note while the rest of the performers continue playing, some notes in an melodic excerpt is played an octave up/down. In these situations, Hough transformation detects either a short, single line segment or several line segments in the region, where a section is being performed. However, as explained in (Section A.2), the synthetic predominant melody do not link to its corresponding location in the performance under these circumstances, unless 70% of the section is covered by the line segments. To handle these problems, a metric, which compensates for octave differences might be devised, analogous to octave-resilient methods used for Eurogenetic musics (Müller et al., 2009). Moreover, arithmetic geometry operations might be made more flexible by removing the 70% coverage constraint and using the ratio of the coverage as a confidence measure for sequential linking. This way, the method will be allowed to link partial similarity between the predominant melodies.

It is also observed that sequential linking predicts a considerable amount of regions which candidate estimate do not (100 vs. 68 false negatives with automatic karar recognition and 75 vs. 39

false negatives with semi-automatic karar recognition). Most of the remaining false negatives (30 false negatives out of 39, and 11 related false positives out of 40 with semi-automatic karar detection) after sequential linking are due to Hough transform not able to yield any links for regions encompassing at least two consequent composition related annotations in the previous step. These regions might be linked to multiple sections by allowing sequential linking make multiple decisions based on the duration of the particular region with respect to the previously linked sections. Nevertheless, the core reason of this type of confusion is due to the partial differences in the predominant melodies explained above. We predict that by implementing the relevant measures proposed above, this type of confusions will diminish without rendering the sequential linking step much more complex.

Another drawback of the method is the detection of the unrelated regions in sequential linking.[4] In this step, unrelated links are currently found indirectly by locating related sections. Even if there are no estimations given for a unrelated region after candidate estimation, sequential linking typically predicts an erroneous link in these regions (16 false positives out of 40 with semi-automatic karar detection), resulting in a low specificity. To increase the detection of true negatives, some direct means of linking the audio signal with some types of unrelated events, i.e. through silence and speech detection, may be useful.

Currently sequential linking does not have any restrictions on the duration of a candidate link. By adding some constraints in the duration of links (i.e. comparison of the performance speed of a candidate in the audio recording by the speed of its synthetic predominant melody and the speeds of the predominant melodies of other sections already linked), an ample amount of erroneous links to silent regions and regions spanning to multiple annotations may also be avoided. Moreover, since the current approach for sequential linking is completely rule-based, every single special case should be considered explicitly, which makes the implementation hard to maintain and prone to errors. This type of situation is highly

[4] Note that candidate estimation does not currently produce any unrelated links since it conceptually only tries to link patterns it is provided, and leaves the time-related decisions to the sequential linking step.

suitable for applying principles of fuzzy logic (Klir & Yuan, 1995). Fuzzy logic might also lower the complexity of the code.

A.7 Conclusion and Future Work

We have proposed a method to link sections of a musical score of a composition with the corresponding regions in an audio recording of the performance of the same composition. We have tested the method with 11 instrumental compositions of OTMM associated with 44 audio recordings, obtaining remarkable performance in a fast operation time.

Since a score section is basically a sequence of note events, the candidate estimation step might be generalized to link any type of melodic fragments with an audio recording. A generalized fragment linking methodology might be helpful in computational tasks such as audio-score alignment, embellishment detection, tonic analysis, tuning detection, intonation analysis and version detection. Conversely, the candidate estimation methodology might require specific adjustments for each task. Comparative candidate estimation experiments should be carried using other techniques such as general Hough transform (Ballard, 1981), SAX (Lin, Keogh, Lonardi, & Chiu, 2003), dynamic programming (Serrà et al., 2009), minimal geodesics (Kimmel & Sethian, 1998).

Currently, candidate estimation uses similarity matrices computed from descriptors which are specifically designed for OTMM. Similarly, the method can be adapted to other musical cultures by computing descriptors, which are musically relevant to the culture being studied.

Appendix B

Applications in Other Music Cultures

While the methodologies presented throughout the thesis are designed to address the culture-specific properties of OTMM, they can be also used to study other musical cultures. This Appendix focuses on three studies: **1)** Score-informed note modeling (explained in Section 6.11), applied to study the tuning and intonation of the svaras of Carnatic music (Section B.1). **2)** Pitch distribution based mode recognition (explained in Section 5.7.2) in Carnatic and Hindustani musics (Section B.7). **3)** Fragment linking (explained in Section 6.3) used to partially align melodic phrases between audio recordings of Cretan music (Section B.8).

In fact, the first two use cases have been initially applied to Carnatic music (Şentürk et al., 2016) and Hindustani music (Chordia & Şentürk, 2013), and later adapted to OTMM. Below, the computational studies are explained in detail.

B.1 Score-Informed Note Models in Carnatic Music

As argued in Section 6.11, defining a musical note as a sound with a definite pitch frequency with possibly minor deviations (e.g. vibratos) may be limited in its purview. To elaborate, we consider the case of Carnatic music, an art music tradition from south India. The counterpart to note in this tradition is referred to as svara, which has a very different musicological formulation. A svara is defined to be a definite pitch value with a range of variability around it owing to the characteristic movements arising from its melodic context. The seven svaras in Carnatic music are $S(a)$, $R(i)$, $G(a)$, $M(a)$, $P(a)$, $D(ha)$, $N(i)$, which account for 12 pitch positions (svarasthanas), S, R_1, $R_2/\ G_1$, R_3/G_2, G_3, M_1, M_2, P, D_1, D_2/N_1, D_3/N_2, N_3 (Krishna & Ishwar, 2012). It is emphasized that the identity of a svara lies in this variability (Krishna & Ishwar, 2012), which makes it evident that the former model of notes has a very limited use in this case. The arguments related to variability are also relevant to Hindustani music, an art music form prevalent in northern parts of the Indian subcontinent and as well as many other melody-dominant music cultures such as Ottoman-Turkish makam music.

In (Şentürk et al., 2016), we discuss a statistical model of notes that broadens the scope of the former, encapsulating the notion of the variability in svaras (Section B.3). We develop a methodology that exploits score information to automatically process the pitch content of audio recordings (Section B.4). The methodology first aligns the audio recording with the relevant music score. This step is designed to handle the structural differences between the music score and the audio performance. Next, the pitch values are aggregated for each note symbol from the aligned instances of the notes and these pitch values are used to compute a statistical representation for each note. The methodology is evaluated extrinsically in a classification task comparing the results with a state-of-the-art system (Koduri et al., 2014) (Section B.5) using two datasets (Section B.2).

Our contributions in (Şentürk et al., 2016) can be summarized as:

1. A novel, computational note model, which is able to describe the characteristics of the notes statistically besides its definite location
2. Adaptation of a state of the art audio-score alignment method proposed for another melody dominant culture to Carnatic music

3. Simplification and generalization of the audio-score alignment method
4. A new dataset of Carnatic music, composed of audio recordings and music scores linked to each other in the document-level

B.2 Data

For evaluation, we use the Carnatic Varṇaṁ test dataset (**CAR-VAR**)[1] (see (Koduri et al., 2014) for a description of varṇaṁs and the dataset). Varṇaṁs are compositions that are often sung to the score unlike several other forms which are interlaced with improvisation. Note that even though the order of the cycles in the score are retained, the performers tend to omit a few cycles or repeat a few of them twice with some minor variations. The dataset has annotations at the metrical cycle-level synchronizing the audio recording and the extracted melody with the score. There are 7 rāgas, 27 recordings and 1155 cycle-level annotations. The average cycle-duration is 9.8 seconds with a standard deviation of 1.2 seconds. The music scores in the dataset are notated as a sequence of svara symbols and their relative durations. The metrical cycles are indicated in the score. There is no nominal tempo information in the score as the performance tempo is decided by the performer. With an assumption that each svara within the cycle is sung exactly according to its relative duration in the score, the svaras in the recording are annotated semi-automatically.

This dataset comes with a limitation that all the performances of a given rāga are of the same composition. Therefore the representation computed for a svara can be specific to either the rāga or the composition. In order to eliminate this ambiguity, we have put together another dataset, called Carnatic Kṛti test dataset (**CAR-KRI**), which is more diverse in terms of the number of compositions per rāga.[2] The details of the dataset are shared in Table B.1. The **CAR-VAR** is drawn from the performances of a compositional form known as varnam. Our dataset contains performances of another compositional form known as kṛti. The latter are more com-

[1] Available at http://compmusic.upf.edu/carnatic-varnam-dataset

[2] The dataset is available at `http://compmusic.upf.edu/node/314`.

Rāga	#Comp.	#Singer	#Rec.
Anandabhairavi	3	5	7
Atana	4	5	5
Bhairavi	5	7	8
Devagandhari	5	5	5
Kalyani	4	4	5
Todi	9	15	15
Total	**30**	**24**	**45**

Table B.1: A more diverse dataset compared to the Carnatic Varṇaṁ test dataset. This consists of 40 recordings in 6 rāgas performed by 24 unique singers encompassing 30 compositions.

mon in concert performances, where the performers take liberty to do an impromptu improvisation. As a result, kṛtis are almost always not sung to the score and hence pose more challenges compared to varṇaṁs for a score-informed approach such as ours. Note that we follow the same format of the scores in **CAR-VAR** to notate the kṛti compositions.

B.3 Model of Musical Notes

Research that involved analysis of svaras in Indian art music has time and again shown that reducing svara to a frequency value results in loss of important information (Subramanian, 2007; Krishnaswamy, 2003; Chordia & Şentürk, 2013). Computational svara descriptions that use more melodic context for the description of a svara such as pitch histograms, have been shown to outperform the naive descriptions such as pitch-class distributions (Koduri et al., 2012, 2014). We build on these observations from the past research and consolidate that to a statistical model of notes that would facilitate extracting information that is otherwise opaque to the currently used model.

Figure B.1 shows melodic contours extracted from the individual recordings of M_1 svara (498 cents in just-intonation) in different rāgas. It shows that a svara is a continuum of varying pitches of different durations, and the same svara is sung differently in

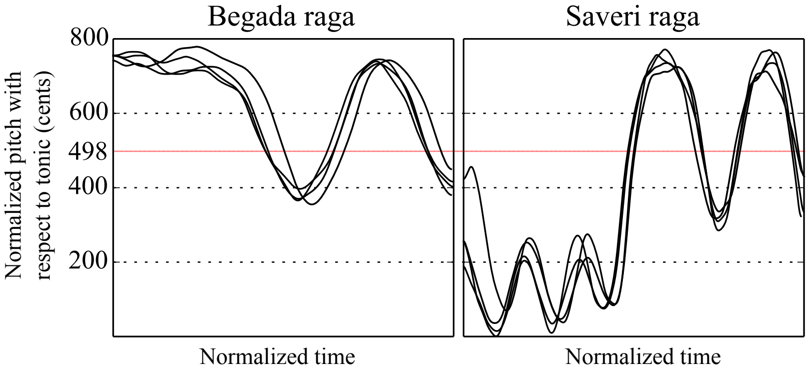


Figure B.1: Example predominant melodies of M_1 svara in different rāgas. the X-axis is time normalized with respect to the length of each predominant melody. The tuning of M_1 svara according to the just-intomation temperament (498 cents) is indicated with a continuous red line. Notice that the majority of the pitches are sung quite distant from the theoretical tuning.

two given rāgas. Note that a svara can vary even within a rāga in its different contexts (Subramanian, 2007; Krishnaswamy, 2003). Taking this into consideration, we propose a statistical model of notes that aims for a more inclusive representation of pitches constituent in a svara. In this model, we define a note as *a probabilistic phenomenon on a frequency spectrum*. This notion can be explored in two approaches that are complementary in nature: i) *temporal*, which helps to understand the evolution of a particular instance of a svara over time (This has been theoretically explored by Krishnaswamy (2003)) and ii) *aggregative*, which allows for studying the whole pitch space of a given svara in its various forms, often discarding the time information.

Our method, presented in the following section, takes the latter approach. From the annotations in our dataset, we aggregate the predominant melodies over the svara reported in Figure B.1 for the same set of rāgas. Figure B.2 shows its representations, computed as described in Section B.4.2. The correspondences between the two figures are quite evident. For instance, M_1 in Bēgaḍa is sung as an oscillation between G_3 (386 cents) and M_1. The representation reflects this with peaks at the corresponding places. Further, the shape of the distributions reflect the nature of pitch activity therein.

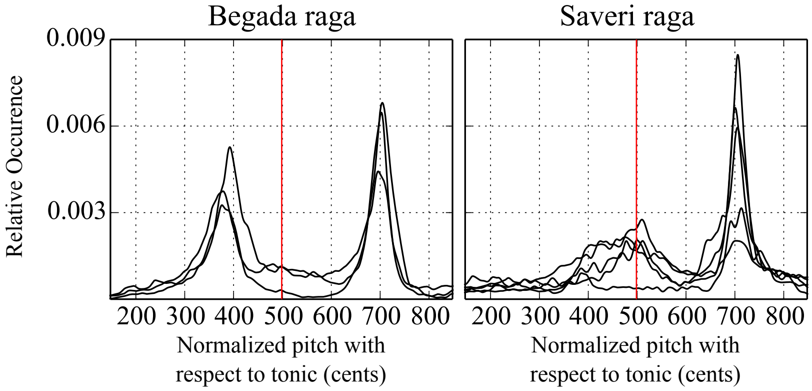


Figure B.2: Histograms of M_1 svara computed from the annotated predominant melodies shown in Figure B.1. The tuning of M_1 svara according to the just-intomation temperament (498 cents) is indicated with continuous red lines.

The goal of our approach is to obtain such representations for svaras across different rāgas in our dataset automatically.

B.4 Methodology

Our method starts by aligning the audio and the score at the cycle- and the svara-level (Section B.4.1). Then the pitch values in different instances of a given svara are obtained and an aggregate representation of a svara is computed (Section B.4.2).

B.4.1 Audio-score alignment

Audio-score alignment can be defined as *the process of finding the segments in the audio recording that correspond to the performance of each musical element in the music score*. For this task, several approaches have been proposed using techniques such as Hidden Markov models (Cont, 2010; Maezawa, Okuno, Ogata, & Goto, 2011), conditional random fields (?, ?) and DTW (Dixon & Widmer, 2005; Fremerey et al., 2010; Niedermayer, 2012).

The structural mismatch between the music score and the audio recording is a typically encountered challenge in audio-score alignment. This is also common phenomenon in the performances of varnams and kritis, where the singers tend to repeat, omit or insert cycles in the score. To overcome this problem there exists methodologies, which allow jumps between structural elements (Fremerey et al., 2010; Holzapfel et al., 2015). However these methodologies are not designed to skip musical events in the performance, which are not indicated in the score, such as impromptu improvisations commonly sung in kritis (Section B.2). Moreover, we may not need a complete alignment between the score and audio recording in order to accumulate a sufficient number of samples for each svara.

In (Şentürk, Holzapfel, & Serra, 2014), an audio-score alignment methodology for aligning audio recordings of OTMM with structural differences and events unrelated to the music score was introduced, and it is later extended to note-level alignment in (Şentürk, Gulati, & Serra, 2014). The methodology proposed in (Şentürk, Holzapfel, & Serra, 2014), divides the score into meaningful structural elements using the editoral section annotations in the score. It extracts a predominant melody from the audio recording and computes a synthetic pitch of each structural element in the score. Then it computes a binarized similarity matrix for each structural element in the score from the predominant melody extracted from the audio recording and the synthetic pitch. The similarity matrix has blobs resembling lines positioned diagonally, indicating candidate alignment paths between the audio and the structural element in the score. Hough transform, a simple and robust line detection method (Ballard, 1981), is used to locate these blobs and candidate time-intervals for where the structural element is performed is estimated. To eliminate erroneous estimations, (Şentürk, Holzapfel, & Serra, 2014) uses a VLMM based scheme, which is trained on structure sequences labeled in annotated recordings. Finally, SDTW is applied to the remaining structural alignments to obtain the note-level alignment (Şentürk, Gulati, & Serra, 2014).

Our alignment methodology is based on the method described in (Şentürk, Holzapfel, & Serra, 2014; Şentürk, Gulati, & Serra, 2014). Since the original methodology is proposed for OTMM, we optimize several parameters according to the characteristics of our data. We also modify several steps in the original methodology for the sake of generalization and simplicity. These changes will be detailed throughout this section, hereafter. The procedure in our methodology can be summarized as:

1. **Feature extraction:** Given an audio recording (a), we extract a predominant melody $\boldsymbol{\varrho}^{(a)}$ using the method proposed in (Salamon & Gómez, 2012). This method has been shown to output reliable pitch estimations on Carnatic music recordings (Koduri et al., 2014). The sampling rate of the predominant melody is equal to ≈ 334.5 Hz, which is reported as an optimal for the methodology in (Salamon & Gómez, 2012). The tonic $\kappa^{(a)}$ is extracted automatically using (Gulati, 2011), which is reported to output near-perfect results in identifying the tonic of Carnatic music recordings. Then, the predominant melody is normalized by tonic frequency and $\hat{\boldsymbol{\varrho}}^{(a)}$ is obtained.
   Parallel to audio predominant melody extraction, the svara symbols notated in the score are mapped to their cent-scale equivalents using just-intonation temperament (Serrà, Koduri, Miron, & Serra, 2011). Then, the score is divided into cycles $\bar{p}_j^{(b)}$ according to the cycle boundaries annotated in the score. For each cycle $\bar{p}_j^{(b)}$, a synthetic pitch $\hat{\boldsymbol{\Psi}}^{\left(\bar{p}_j^{(b)}\right)}$ is computed by sampling a hypothetical continuous predominant melody corresponding to the svara sequence (Şentürk, Holzapfel, & Serra, 2014) (Section 4.2.2). In this process, the tempo $\tau^{(b)}$ of the score is considered as 70 bpm, which is reported in (Koduri et al., 2012) as the average tempo in the **CAR-VAR**. The sampling rate of the synthetic pitch is equal to the sampling frequency of the audio predominant melody. During the synthetic pitch computation, the svara onset and offset timestamps $t\left(\bar{n}_k^{(b)}\right) \mid \bar{n}_k^{(b)} \in \bar{\mathbf{N}}^{(b)}$ are recorded. This information is used to obtain the svara-level alignment later.
2. **Cycle-Level Alignment:** Instead of Hough transform used in (Şentürk, Holzapfel, & Serra, 2014), we use `ISDTW` (explained in Section 6.3.2), a common methodology used to find a queried subsequence in a given target (Müller & Appelt, 2008; Anguera & Ferrarons, 2013) to estimate the time-intervals, where a cycle is performed. Our preliminary experiments on the Carnatic Varnam Dataset showed that using `ISDTW` gave comparable results to Hough transform. More-

over, ISDTW simplifies the note-level alignment step compared to (Şentürk, Gulati, & Serra, 2014) since note onset and offsets can be directly inferred from the paths obtained from ISDTW, without introducing an additional process (e.g. SDTW in (Şentürk, Gulati, & Serra, 2014) as described in Section 6.3.2). We set the step size to $\{(2,1),(1,1),(1,2)\}$. This step size restricts the path between half and double of the tempo, which helps to avoid pathological errors. To obtain an accumulated cost matrix, $\boldsymbol{A}^{\kappa^{(a)},(a,\bar{p}_j^{(b)})}$ for each cycle $(\bar{p}_j^{(b)})$ (Equation 6.6), we use the local distance measure defined in Equation 5.3. Remember that this distance may be interpreted as the shortest distance in cents between two pitch classes and it is not affected by octave-errors in the normalized predominant melody.

We use iterative subsequence dynamic time warping (ISDTW) given in (Müller, 2007, Page 81) (Section 6.3.2) to estimate multiple alignments for each cycle $\left(\bar{p}_j^{(b)}\right)$. We iterate the algorithm for 12 times for each cycle. After each iteration, we obtain a set of cycle estimations $\left\{\bar{p}_i^{(a)} : p_i^{(a)} = p_j^{(b)}\right\}$ for each cycle $\left(\bar{p}_j^{(b)}\right)$ and each estimation has an optimal alignment path, $\boldsymbol{\varpi}\left(\bar{p}_i^{(a)}, \bar{p}_j^{(b)}\right) = \left[\varpi_1\left(\bar{p}_i^{(a)}, \bar{p}_j^{(b)}\right), \varpi_2\left(\bar{p}_i^{(a)}, \bar{p}_j^{(b)}\right), \ldots\right] \mid p_i^{(a)} = p_j^{(b)}$, a time-interval $t\left(\bar{p}_i^{(a)}\right)$ and a similarity-value $\nu\left(\bar{p}_i^{(a)}, \bar{p}_j^{(b)}\right)$ (computed using Equation 6.8). The l^{th} point in the path is expressed as $\varpi_l\left(\bar{p}_i^{(a)}, \bar{p}_j^{(b)}\right) = \left(r_l\left(\bar{p}_i^{(a)}\right), q_l\left(\bar{p}_j^{(b)}\right)\right)$, where $r_l\left(\bar{p}_m^{(a)}\right)$ and $q_l\left(\bar{p}_j^{(b)}\right)$ refer to the sample indices in the predominant melody of the audio recording (a) and the synthetic melody of the score cycle $\left(\bar{p}^{(b)}\right)$, respectively. Section B.5 presents the experiments to search the optimal value for the binarization threshold, $\beta(\boldsymbol{B})$, used in the similarity computation. After each iteration, the indices between $r_l\left(\bar{p}_i^{(a)}\right) \pm 0.1 \times \left|\hat{\boldsymbol{\Psi}}^{\left(\bar{p}_j^{(b)}\right)}\right|$ in the accumulated cost matrix, $\boldsymbol{A}^{\kappa^{(a)},\left(a,\bar{p}_j^{(b)}\right)}$, are set to infinity so that a new path will not be searched nearby in the next iteration.

3. **Discarding erroneous estimations:** At this step a considerable number of correct estimations are obtained albeit with a comparable number of erroneous estimations. Nonetheless, a high precision has to be ensured in the cycle-level alignment to obtain a reliable svara description. In order to achieve this, a trade in the recall can be afforded in the process since a moderate recall in the cycle-level alignment would still be able to supply a good number of samples per svara.

   The method proposed for discarding erroneous estimations in (Şentürk, Holzapfel, & Serra, 2014) is not generalizable as introducing a new form with a different structure requires substantial number of training recordings in that form. For this reason, we choose to use an unsupervised estimation selection scheme, which is more generalizable and simpler.

   The estimated cycles are clustered into two classes with respect to their similarity values (i.e. “good” and “bad” estimations). k-means clustering (MacKay, 2003) is used to cluster the estimations. We use squared Euclidean distance as the distance measure and discard the cluster with low scores. Next, the estimations, which overlap more than 3 seconds in time, are grouped. In each group, only the estimation with the highest similarity-value is kept as the music has a single melody track throughout. In Section B.5, the alignment results after discarding estimations both without (i.e. only discarding overlapping estimates) and with k-means clustering are reported.

4. **Note-Level Alignment:** Recall that the svara onset timestamp, $t_{ini}\left(\bar{n}_k^{(b)}\right)$, and offset timestamp, $t_{fin}\left(\bar{n}_k^{(b)}\right)$, in each cycle of the synthetic pitch, $\hat{\boldsymbol{\Psi}}^{\left(\bar{p}_j^{(b)}\right)}$, are known. The aligned svara onset and offsets are directly obtained as the timestamps, $t\left(r_l^{\left(\bar{p}_i^{(a)}\right)}\right)$, which are mapped to these onsets and offsets inside the alignment path, $\boldsymbol{\varpi}\left(\bar{p}_i^{(a)}, \bar{p}_j^{(b)}\right)$.

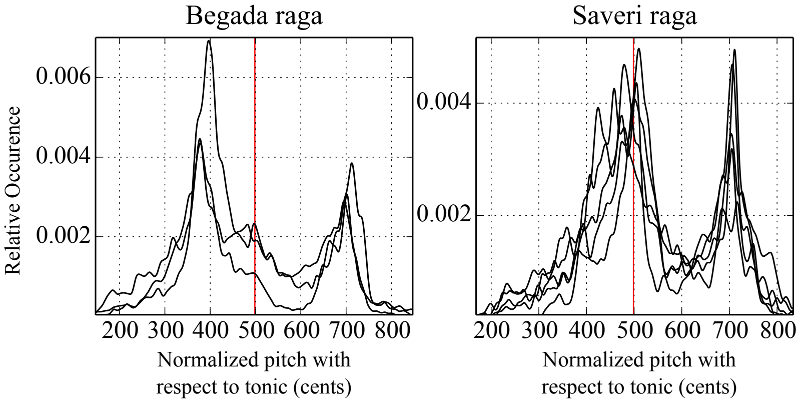


Figure B.3: Description of M_1 svara (498 cents in just intonation) using our approach.

B.4.2 Computing svara representations

For a given recording, for each svara, n_k, in the corresponding rāga, we obtain a pool of normalized pitch values, $\left\{\hat{\rho}_1^{(n_k)}, \hat{\rho}_2^{(n_k)}, \ldots\right\}$, aggregated over all the aligned instances from its melodic contour. Our representation must capture the probabilities of the pitch values in a given svara. Histograms are a convenient way for representing the probability density estimates (Chordia & Şentürk, 2013; Koduri et al., 2014). Therefore, we compute a normalized histogram $\hat{\boldsymbol{H}}^{(n_k)}$ over the pool of the pitch values as described in Section 5.5, Equation 5.4. For brevity sake, we consider pitch values over the middle octave (i.e., starting from the tonic) at a bin-resolution $b\left(\hat{\boldsymbol{H}}\right)$ of one cent.

Figure B.3 shows the representations obtained in this manner for M_1 svara (our running example from Figure B.1) in different rāgas. Notice that the representations obtained for M_1 are similar to the corresponding representations shown in Figure B.2. This representation allows to deduce important characteristics of a svara besides its definite location (i.e., 498 cents) in the frequency spectrum. For instance, from Figure B.3, one can infer that M_1 in Begada and Saveri are sung with an oscillation that ranges from G_3 (386 cents) to P (701 cents) in the former and M_1 to P in the latter.

B.5 Evaluation and Results

Our method is evaluated on the two datasets described in Section B.2 using the following tasks:

i. The cycle-level alignment, evaluated intrinsically using the ground truth annotations from the Carnatic Varṇaṁ test dataset.
ii. The svara-level alignment and the computed representation, evaluated extrinsically on both Carnatic Varṇaṁ test dataset and the Carnatic Kṛti test dataset via rāga classification task.

The svara-level alignments cannot be verified in an intrinsic manner because marking the ground truth is prone to be erroneous as it is difficult for even musicians to agree with each other on the exact boundaries of a svara sung in a melodic continuum.[3]

To evaluate the cycle-level alignment, we check the time-distance between the estimated borders of the cycle and annotated borders as described in (Şentürk, Holzapfel, & Serra, 2014) (Section 6.7.4). A cycle is marked as a true positive if the distance between both of the boundaries of the aligned cycle and the relevant annotation is less than 3 seconds. It is marked as a false positive otherwise. If there is no estimation for an annotation, it is marked as a false negative.

Figure B.4 shows the recall, precision and F_1-score for different binarization thresholds used in similarity computation. Figure B.4a shows that our methodology achieves a balanced recall and precision in the cycle-level alignment even without having a precise information on the performance tempo. Figure B.4b shows that the process described to the discard erroneous alignments removes most of the false positives within an acceptable decrease in recall. It can also be observed that our cycle-level alignment is insensitive to the binarization threshold, $\beta(\boldsymbol{B})$. When the parameter is selected between 50 cents (a quarter tone) and 200 cents (a whole tone), there is no a significant difference in the alignment results at the $p = 0.01$ level as determined by a multiple comparison test

[3] The experiments and the results are available at http://compmusic.upf.edu/node/314.

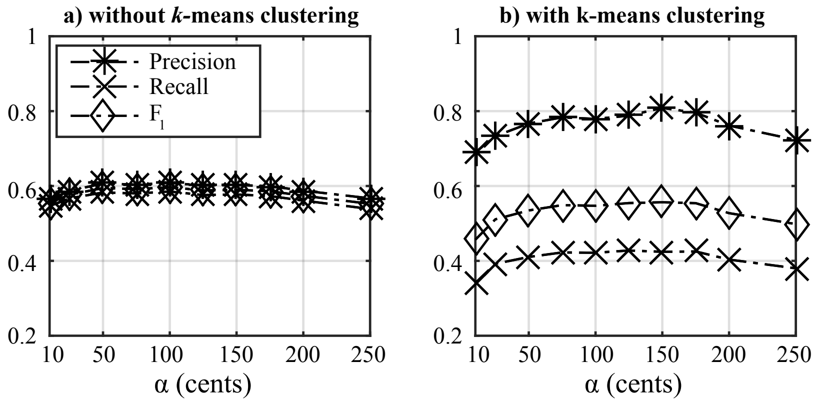


Figure B.4: Results of cycle-level alignment for different binarization threshold values.

Method	CAR-VAR	CAR-KRI
Context-based svara distributions (Koduri et al., 2014)	0.62	0.64
Our approach	0.95 to 1	0.88
Using the groundtruth annotations	0.95	N/A

Table B.2: Results of rāga classification task over the two datasets using different approaches.

using the Tukey-Kramer statistic. Hereafter, we report results for a binarization threshold of 150 cents.

Using an $\beta(\boldsymbol{B})$ of 150 cents, we achieve a 0.42 recall, 0.81precision and 0.56 F_1-score in cycle-level alignment after discarding the erroneous estimations. The mean and the standard deviation of the true positives are 0.62 and 0.59 seconds, respectively. Within the Carnatic Varnam dataset, we align 606 cycles and 15795 svaras in total. Out of these cycles 490 are true positives. By inspecting the false positives we observed two interesting cases: occasionally an estimated cycle is marked as false positive when one of the boundary distances is slightly more than 3 seconds. The second case is when the melody of the aligned cycle and performance is similar to each other, e.g. $\nu\left(\bar{p}_i^{(a)}, \bar{p}_j^{(b)}\right) > 0.6$. In both situations considerable number of the note-level alignments would still be useful for the svara model. Within **CAR-KRI**, 1938 cycles and 59209 svaras are aligned in total.

We use a rāga classification task to evaluate the correctness of the svara alignments and the usefulness of the svara representation created using our statistical model. Our svara representation was shown to perform better compared to the existing representations in our previous work (Koduri et al., 2014). Therefore, in this task our primary motive is to evaluate the correctness of the svara alignments. However, as marking the svara boundaries is not a viable task, we combine it with evaluating the usefulness of the representation itself in a rāga classification task. We parametrize the representation of each svara using a set of features proposed in our aforementioned work, which include salient observations and the shape parameters of the histogram:

i. The highest probability value in the histogram of the svara
ii. The pitch value corresponding to the highest probability
iii. A probability-weighted mean of pitch values
iv. Pearson's second skewness coefficient
v. Fisher's kurtosis
vi. Variance

There are 12 svaras in Carnatic music, where each rāga has a subset of them. For the svaras absent in a given rāga, we set the features to a nonsensical value. Each recording therefore has 72 features in total.

The smallest rāga-class has three recordings in the Carnatic Varnam dataset, with few classes having more, so we subsampled the dataset six times (corresponding to the highest number of recordings for a class) with three recordings per class. We have also subsampled our dataset in a similar manner. The k-nearest neighbors classifier was earlier shown to perform the best in several rāga classification tasks with varied feature sets (Koduri et al., 2014). We use the same, with Euclidean distance metric and the number of neighbors set to one.

We compare the results of our approach with the one proposed by Koduri et al. (2014) which was shown to outperform the previous methods of rāga classification by a slight margin. Their approach uses a moving window to estimate the local temporal context of a small section of melodic contour which is further used to estimate the svara sung at that instance. For each svara, we obtain the corresponding pitch values and use them to create a representation using the method described in Section B.4.2, and parametrize it as described earlier in this section. We further compare these results with that obtained using the representation computed from the annotated svara instances in the dataset.

We performed the classification experiment over the subsamples of the two datasets using the leave-one-out cross-validation technique. For our approach, we repeated the experiment with the alignment data resulting from different binarization thresholds. The mean F_1-scores using the representations obtained from the annotations in the dataset, our approach and (Koduri et al., 2014) across the subsampled datasets for the two datasets are reported in Table. B.2. Our approach has performed significantly better than the earlier one in (Koduri et al., 2014) on both datasets, and is on par with the method using annotated data. This is a strong indication that our description using the statistical model succeeds in capturing the variability, and therefore the identity of svaras. We also observed that different binarization threshold values have a unimportant impact on the classification accuracy.

B.6 Summary

We have presented a statistical model of musical notes that expands the scope of the current model in use by addressing the notion of variability of svaras. An approach that builds on this model and exploits scores to describe pitch content in the audio music recordings is presented and evaluated at various levels. The results clearly indicate that our approach is successful in obtaining a computational description of the svaras improving over the state-of-the-art results significantly.

CAR-VAR has 7 rāgas, one composition per rāga sung by 3 to 5 artists. We believe this to be one of the contributing factors to a near perfect result using our approach in the rāga classification test. We have put together a more diverse dataset that encompasses more compositions per rāga. Our approach has been shown to be robust to the variability of svaras across compositions in a given rāga. However, we seek attention to the fact that our alignment method relies on the average tempo of the recordings computed from the annotations of the Carnatic Varnam dataset (Koduri et al., 2014). In order to make the system more self-reliant, we plan to add an initial tempo estimation step similar to (Holzapfel et al., 2015) by aligning a single cycle using SDTW and resynthesizing the synthetic melodies according to the estimated tempo. We also plan to improve the alignment step by incorporating the svara models in the similarity computation within a feedback mechanism.

An interesting direction to our work is to infer possible facts about a svara from its description. For instance, answering questions such as: **i)** "Is the svara sung steadily?" **ii)** "Where is the oscillation on a svara anchored?" and so on. These can further be used as parameters that describe the svara even more concisely. Another direction which interests us is the development of alternative computational descriptions using our statistical model of notes.

B.7 Mode Recognition in Carnatic and Hindustani Music

The multi-distribution per rāga method (Chordia & Şentürk, 2013) implemented in **MO**de **R**ecognition and **T**onic **Y**dentification Toolbox (Karakurt et al., 2016) (MORTY) (Karakurt et al., 2016) (described in Section 5.7.2) has been used as a benchmark for rāga/rāg recognition of audio recordings of Hindustani and Carnatic music in comparison with two novel methods (Gulati, Serrà, Ganguli, et al., 2016; Gulati, Serrà, Ishwar, et al., 2016). In both papers, the optimal parameters of the multi-distribution per rāga method reported in (Chordia & Şentürk, 2013) are used (bin size $b\left(\hat{\boldsymbol{H}}\right) = 10$ cents, kernel width $\sigma\left(\hat{\boldsymbol{H}}\right) = 10$ cents, number of nearest neighbors $k = 1$ using Bhattacharyya distance). Note that there already exists a method for tonic identification for these music traditions (Salamon, Gulati, & Serra, 2012; Gulati, 2011), which is reported to provide near perfect results. This method is used in both papers to automatically identify the tonic. Therefore, the tonic identification and joint estimation experiments are not conducted. For more details on the proposed methodologies and experiments, please refer to (Chordia & Şentürk, 2013; Gulati, Serrà, Ishwar, et al., 2016) and (Gulati, 2016, Chapter 6).

Table B.3: The highest accuracies obtained by the methods compared in (Gulati, Serrà, Ishwar, et al., 2016).

	(Gulati, Serrà, Ishwar, et al., 2016)	(Chordia & Şentürk, 2013)	(Koduri et al., 2014)
10 rāga dataset	91.7%	89.5%	70.1%
40 rāga dataset	69.6%	74.1%	51.4%

In (Gulati, Serrà, Ishwar, et al., 2016) a rāga recognition method for Carnatic music based on melodic motif characterization using graph analysis is proposed. The method has been compared with the multi-distribution per rāga method (Chordia & Şentürk, 2013) and a method based on parameterizing the pitch distributions computed from individual svaras (Koduri et al., 2014). The methods are evaluated two datasets with 10 rāga and 40 rāga setups in a 10-fold cross validation scheme. Table B.3 shows the accuracies obtained by the aforementioned methodologies on these datasets. The methodologies proposed in (Gulati, Serrà, Ishwar, et al., 2016) and (Chordia & Şentürk, 2013) output similar results and they significantly outperform the method proposed in (Koduri et al., 2014). When the confusions were inspected, it was seen that the methods proposed by Gulati, Serrà, Ishwar, et al. (2016) and (Chordia & Şentürk, 2013) provided complementary results. Gulati, Serrà, Ishwar, et al. (2016) is better in distinguishing "allied" rāgas, which have similar scales but different melodic progressions, while the multi-distribution per rāga method (Chordia & Şentürk, 2013) is more successful in recognizing scale-based rāgas.

In the latter work (Gulati, Serrà, Ganguli, et al., 2016), a method based on extracting a novel feature termed by the authors as the "time-delayed melody surface" is proposed for rāga/rāg recognition in Carnatic and Hindustani music. This method is compared against the multi-distribution per rāga method (Chordia & Şentürk, 2013) and the method previously proposed in (Gulati, Serrà, Ishwar, et al., 2016) on the same 40 rāga dataset used in (Gulati, Serrà, Ishwar, et al., 2016) and a new 30 rāg dataset of Hindustani music. In the experiments, leave-one-out cross validation is used instead of 10-fold cross validation. The comparative results are summa-

Table B.4: The highest accuracies obtained by the methods compared in (Gulati, Serrà, Ganguli, et al., 2016).

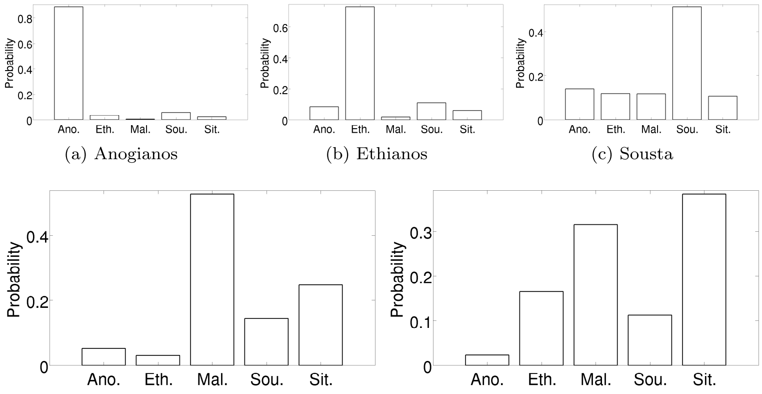
	(Gulati, Serrà, Ganguli, et al., 2016)	(Chordia & Şentürk, 2013)	(Gulati, Serrà, Ishwar, et al., 2016)
40 rāga dataset	97.7%	91.7%	83.0%
30 rāg dataset	86.6%	73.1%	68.1%

rized in Table B.4. On both datasets, the multi-distribution per rāga method (Chordia & Şentürk, 2013) significantly outperforms the method previously proposed in (Gulati, Serrà, Ishwar, et al., 2016). However, the method proposed by Gulati, Serrà, Ganguli, et al. (2016) significantly outperforms both methods. The results show that time-delayed melody surface is a superior feature than pitch distribution on the rāga/rāg recognition task and potentially in mode recognition tasks in other music cultures.

B.8 Melodic Phrase Matching in Cretan Music

Recently, the characteristics of the leaping dances of Crete has been studied by Holzapfel (in press, 2015a). The author focuses on identifying the melodic key phrases, which occur across the audio recordings of different dances. The audio recordings are selected from the commercial recordings of well-known performers. Initially, the author applies automatic beat detection to the audio recordings using the methodology proposed in (Davies & Plumbley, 2007) and then corrects the errors manually. From the beats the measure boundaries are inferred directly from the knowledge that the studied dances follow a $2/4$ musical meter.

Next, the predominant melody is extracted from each recording using MELODIA (Salamon & Gómez, 2012). The extracted predominant melody is sliced into patterns of two measures long with an overlap of a measure. Each pattern in a recording is searched in the rest of the recordings using the fragment linking method proposed in (Şentürk, Holzapfel, & Serra, 2014) (Section 6.3). Af-



(a) Anogianos (b) Ethianos (c) Sousta

(d) Maleviziotis (e) Sitiakos

Figure B.5: Distribution of the detected melodic patterns for the five leaping dances of Crete. The Figure is reproduced from (Holzapfel, in press), courtesy of André Holzapfel.

ter matching the patterns by this partial audio-to-audio alignment scheme, the author reports the probability of a pattern played in a particular dance observed in other dances (or itself) (Figure B.5). From the probability distributions, the author discusses the relations between different dances in terms of their geographical origins, performance practice and similarity of movements.

B.9 Conclusion

This Appendix presented examples of how several analysis methods described throughout the thesis are used in other music cultures. In the future, I would like to extend the methodology to automatically adapt the parameters according to the culture-specific properties of the studied music culture.

Note that the developed methodologies are not necessarily limited to melody-dominant music traditions. As an example, semi-improvised jazz music performances, where musicians build variations of predefined melodies through improvisation, share a similar basis with OTMM. The variations of a characteristic motif may be identified through out a performance using the fragment linking methodology described in Section 6.3 similar to (Holzapfel, 2015a). In this case, computing HPCPs (Section 5.3) from the notated melody (based on jazz harmony) might be more suitable than computing a monophonic predominant melody. Similarly, audio-score alignment proposed in Chapter 6 might be adapted to structure analysis in Eurogenetic musics by replacing the predominant melodies with some harmonic descriptors.

Appendix C

Towards Open and Reproducible Research

In the start of the CompMusic project, there had been a lack of open-source software tools, which aimed at computational analysis of the studied music cultures. Moreover, most of the existing automatic analysis tools such as Essentia (Bogdanov et al., 2013) and TarsosDSP (Six et al., 2014) could be mostly utilized for extracting low-level features without any culture-aware processing and therefore these tools were not able to address most of the relevant research problems. While Makam Toolbox by Bozkurt (2008) has been an invaluable tool for automatic description of OTMM recordings, the toolbox is not practical to extend and deploy (e.g. to Dunya-makam) due to its implementation in MATLAB, a proprietary programming language. Likewise, the lack of music corpora representing the studied music traditions (Chapter 3), test datasets tailored for the studied research tasks (Section 3.2) as well as reproducible experiments and the obtained results.

Initially, I had not identified the lack of public tools, datasets and experiments as a problem itself. As a result, many of the early research presented in this thesis (such as (Şentürk et al., 2012)) are not reproducible. In later stages of my doctoral research, I experienced the absence of resources and tools as a major obstacle for reproducing, building on top of and hence advancing music information research (MIR). Therefore, I have dedicated a fair amount of effort in the last year of my PhD to reimplement the existing state-of-the-art applied on OTMM (e.g. most of the methodologies described in Chapter 5) and package the implementations of the novel methodologies proposed in this thesis (e.g. the audio-score alignment methodologies described in Chapter 6) with modularity, readability and extensibility in mind. In the meantime, I have either spearheaded or contributed to the creation, curation and maintenance of the collections Dunya-makam corpus and the test datasets described in Chapter 3. Moreover, I share the results of almost all of the experiments done throughout my doctoral research with the extracted features and complementary metadata. Some of my later publications, e.g. (Şentürk & Serra, 2016b; Karakurt et al., 2016) are designed to be fully reproducible.

The rest of the Appendix is organized as follows: Section C.1 presents the open tools, coding practices and the services used. Section C.2 explains the organization of the test datasets, experiments and publications. Section C.3 finalizes this Appendix with a brief conclusion.

C.1 Software

The analysis tasks described throughout the thesis are packaged into separate repositories for the sake of modularity. They are publicly hosted in *GitHub*[1] with complete version history. To promote open science and reproducibility, the repositories are published free of charge under the AGPLv3.[2] Within the development cycle, the code is periodically released according to the conventions defined by the *Semantic Versioning v*2.0.0.[3] Moreover, each release is automatically assigned a digital object identifier (DOI) using *Zenodo*'s[4] Github integration to be able to discover and cite the code with proper versioning.[5] In order to detect erroneous behaviour (e.g. bugs) immediately and ensure reliable operation, the majority of the repositories consist of unit tests. These unit tests,

[1] https://github.com/

[2] https://www.gnu.org/licenses/agpl-3.0.en.html

[3] http://semver.org/spec/v2.0.0.html

[4] https://zenodo.org/

[5] https://guides.github.com/activities/citable-code/

along with code style validation,[6] code coverage[7] and code quality[8] checkers, are invoked automatically using Travis CI,[9] when a "commit" is pushed to GitHub. This automated setup, helps to identify and solve software bugs earlier; reduces the time, effort and risks introduced by code repetition; improves readability and enhances sustainable development, in general.

Except the audio-score alignment code, all tools are implemented in Python 2.7. They generally follow the conventions of object-oriented programming. For coding style consistency, *PEP* 8 style guide[10] is strictly followed. The code in each package is organized into modules.[11] Moreover the packages include the setup scripts,[12] which would allow the user to install these packages to different machines with ease. The code depends on other open source software such as *NumPy* (van der Walt, Colbert, & Varoquaux, 2011) and *SciPy* (Jones, Oliphant, Peterson, et al., 2001–) for numeric computations, *scikit-learn* (Pedregosa et al., 2011) for machine learning related tasks, *NetworkX* (Hagberg, Schult, & Swart, 2008) for graphs analysis, *pandas* (McKinney, 2010) for processing tabular data, *matplotlib* (Hunter, 2007) for visualizations, *MusicBrainz NGS bindings*[13] for crawling MusicBrainz, LilyPond (Nienhuys & Nieuwenhuizen, 2003) for score engraving, Essentia (Bogdanov et al., 2013) for audio processing and *eyeD3*[14] for reading embedded metadata in audio files. *Jupyter notebooks*[15] are provided as "user manuals" of each package to demonstrate example usage. Since one of motivations of the thesis is handling large digital audio collections, we also provide examples of parallelization through *ipyparallel*,[16] which is a part of the *IPython*

[6] using *flake8* (`http://flake8.pycqa.org/en/latest/`) for the code in Python

[7] using *codecov* (`https://codecov.io/`) for the code in Python

[8] using *QuantifiedCode* `http://docs.quantifiedcode.com/` for the code in Python

[9] `https://travis-ci.org/`

[10] `https://www.python.org/dev/peps/pep-0008/`

[11] `https://docs.python.org/2/tutorial/modules.html`

[12] `https://docs.python.org/2/distutils/setupscript.html`

[13] `https://github.com/alastair/python-musicbrainzngs`

[14] `http://eyed3.nicfit.net/`

[15] `http://jupyter.org/`

[16] `https://github.com/ipython/ipyparallel/releases`

project (Pérez & Granger, 2007).

Two repositories are not written in Python. The symbolic phrase segmentation methodology (explained in Section 4.4) is implemented in MATLAB scripting language.[17] The repository is a fork of the original source code by Bozkurt, Karaosmanoğlu, et al. (2014). The fork introduces performance optimizations and wrapper functions for several steps such as feature extraction, training and testing. The second is the audio-score alignment code, which is mainly written in MATLAB scripting, except the `DTW` implementation in C. The `DTW` implementation is primarily written by Sankalp Gulati[18] with modifications for OTMM (such as the implementation of Equation 5.3) introduced by myself. The implementation is compiled as a *mex* function[19] to interface with the MATLAB scripts. Currently, the phrase segmentation[20] and audio-score alignment[21] code is compiled into binaries for Linux and MacOSX using MATLAB compiler[22] in MATLAB R2015a (8.5), hence the algorithms can be called using MATLAB Runtime [23] without the need of a MATLAB proprietary license. In the future, we would like to port the audio-score alignment code to *Cython*[24] (i.e. the `DTW` implementation will stay in C). We would also like to update all the code in Python 2.7 to latest version of Python 3 for sustainability in the future.[25]

To easily analyze large-scale audio recording and music score collections of OTMM, I have implemented a toolbox called **Turk**ish-**O**ttoman **M**akam (M)usic **A**nalysis **TO**olbox (`tomato`). `tomato` is a comprehensive and easy-to-use, which implements the state-of-the-art methodologies (explained throughout the thesis) designed specifically for the culture-specific characteristics of OTMM. `tomato` calls the score metadata extraction and structure analysis (Sec-

[17] `http://es.mathworks.com/products/matlab/`

[18] The implementation is included in the repository `https://github.com/sankalpg/Library_PythonNew`

[19] `http://es.mathworks.com/help/matlab/ref/mex.html`

[20] Hosted in its package releases: `https://github.com/MTG/makam-symbolic-phrase-segmentation/releases`

[21] Hosted in `https://github.com/sertansenturk/tomato_binaries`

[22] `http://www.mathworks.com/products/compiler/`

[23] `http://www.mathworks.com/products/compiler/mcr/`

[24] `http://cython.org/`

[25] We will start working on *Python* 3+ support, as soon as the Essentia bindings are available: `https://github.com/MTG/essentia/issues/138`.

tion 4.4) and each step in complete audio analysis (Section 5.11) and complete joint analysis (Section 6.12). The toolbox is designed such that the complete analysis methods are able to output partial results in case some steps fail. As described in Section 7.1, `tomato` is already integrated to Dunya-makam, the prototype web application of CompMusic for the discovery of OTMM. `tomato` also includes pretrained models for score phrase segmentation and audio makam recognition. These models as well as any other type of data (the music scores, extracted features, figures, outputs etc.) in the repository are licensed under the *Creative Commons Attribution-NonCommercial-ShareAlike 4.0 International License*.

The implementations of all methods described in the Chapters 4-6 are summarized in the Tables C.1-C.3. In the future, we would like to port all the repositories inside `tomato` to get rid of the circular-dependencies across the packages, provide a more comprehensive documentation and extend the unit tests. We would also like to create a Docker[26] image consisting of `tomato` not only to deploy the toolbox to web services easily but also to improve the reproducibility of the software environment.

C.2 Data, Experiments and Publications

Similar to the software tools, the **SymbTr** collection (Section 3.1.2) and all of the test datasets (Section 3.2) are hosted in Github. Using git allows us to track the changes in the data easily, with complete version history.[27]

When a test dataset is used in a publication, a new version is released with proper tagging.[28] Likewise, a new version of the **SymbTr** collection is released,[29] when certain milestones are met.[30] Similar to the unit tests applied to the code, the **SymbTr** collec-

[26] `https://hub.docker.com/`

[27] As an example, the readers can inspect `https://github.com/MTG/SymbTr/commits/v2.4.3/txt/rast--sarki--sofyan--gelmez_oldu--dramali_hasan_hasguler.txt` and `https://github.com/MTG/SymbTr/blame/v2.4.3/txt/rast--sarki--sofyan--gelmez_oldu--dramali_hasan_hasguler.txt`.

[28] e.g. `https://github.com/MTG/otmm_tonic_dataset/releases`

[29] `https://github.com/MTG/SymbTr/releases`

[30] `https://github.com/MTG/SymbTr/blob/v2.4.3/Changelog.md`

Table C.1: An overview of the implementations of the score analysis methodologies and score format converters explained in Chapter 4.

Task	Methodology	Inputs	Package	Version	Comments
Metadata extraction	Section 4.1	Work MBID	`https://github.com/sertansenturk/makammusicbrainz`	v1.3.0	Requires internet connection to access MusicBrainz
Phrase segmentation	(Bozkurt, Karaosmanoğlu, et al., 2014), Section 4.3.2	**SymbTr**-txt score	`https://github.com/MTG/makam-symbolic-phrase-segmentation`	v1.0-alpha.1	Forked from `http://akademik.bahcesehir.edu.tr/~bbozkurt/112E162.html`. The binary is hosted in the releases of the package, called *phraseSeg*
Section extraction	(Şentürk & Serra, 2016b), Section 4.3.2	**SymbTr**-txt score	`https://github.com/sertansenturk/symbtrdataextractor`	v2.1.0	Implemented in *symbtrdataextractor/section.py*
Semiotic labeling	(Şentürk & Serra, 2016b), Section 4.3.2	**SymbTr**-txt score	`https://github.com/sertansenturk/symbtrdataextractor`	v2.1.0	Implemented in *symbtrdataextractor/structurelabeler.py*
Synthetic melody extraction	(Şentürk, Holzapfel, & Serra, 2014), Section 4.2.2	**SymbTr**-txt score	`https://github.com/sertansenturk/symbtrdataextractor`	v2.1.0	Implemented in *symbtrdataextractor/scoreprocessor.py*. *fragmentLinker* package (Table C.3) uses an internal implementation.)
Complete score analysis	Sections 4.1- 4.3	**SymbTr**-txt score, **SymbTr**-mu2 score (optional), work MBID (optional)	`https://github.com/sertansenturk/symbtrdataextractor`	v2.1.0	Calls all the tasks above and also cross-validates the metadata obtained from MusicBrainz, **SymbTr**-txt and **SymbTr**-mu2 scores
SymbTr-txt to MusicXML conversion	Section 4.4.2	**SymbTr**-txt score, **SymbTr**-mu2 score (optional), work MBID (optional)	`https://github.com/burakuyar/MusicXMLConverter`	v1.2.1	**SymbTr**-mu2 and work glsMBID can be input to enhance the work metadata embedded to the MusicXML score
SymbTr-MusicXML to LilyPond conversion	Section 4.4.2	**SymbTr**-MusicXML score	`https://github.com/hsercanatli/makam-musicxml2lilypond`	v1.2.1	
LilyPond to SVG conversion	Section 4.4.2	**SymbTr**-MusicXML score	`https://github.com/sertansenturk/tomato`	v0.9.1	Implemented in *tomato/symbolic/scoreconverter.py*
Complete score analysis and format conversion	Chapter 4	**SymbTr**-txt score, **SymbTr**-mu2 score (optional), work MBID (optional)	`https://github.com/sertansenturk/tomato`	v0.9.1	Implemented in *tomato/symbolic*. Depends on all packages above.
Additional **SymbTr** tools	Section 4.4.3	**SymbTr**-txt or **SymbTr**-mu2 score	`https://github.com/MTG/SymbTr-extras`	v0.3.dev	Tools to manipulate the music scores to maintain consistency in formatting (e.g. encoding, line breaks), content (e.g. rest names, usul annotations) etc. Depends on *makammusicbrainz, symbtrdataextractor* and *MusicXMLConverter*.

Table C.2: An overview of the implementations of the audio analysis methodologies explained in Chapter 5.

Task	Methodology	Inputs	Package	Version	Comments
Metadata extraction	Section 5.1	Audio file or recording MBID	`https://github.com/sertansenturk/makammusicbrainz`	v1.3.0	Requires internet connection to access MusicBrainz
Predominant melody extraction	(Atlı et al., 2014), Section 5.2.1	Audio file	`https://github.com/sertansenturk/predominantmelodymakam`	v1.2.0	See `ATL-MEL`
Predominant melody filtering	(Bozkurt, 2008), Section 5.2.1	Predominant Melody	`https://github.com/hsercanatli/pitchfilter`	v1.2.1	See $\texttt{ATL-MEL}_f$
Pitch distribution extraction	(Bozkurt, 2008), Section 5.5	Filtered predominant melody (using $\texttt{ATL-MEL}_f$)	`https://github.com/altugkarakurt/morty`	v1.2.1	Implemented in *morty/pitchdistribution.py*
Pitch-class distribution extraction	(Chordia & Şentürk, 2013), Section 5.5	Filtered predominant melody (using $\texttt{ATL-MEL}_f$)	`https://github.com/altugkarakurt/morty`	v1.2.1	Implemented in *morty/pitchdistribution.py*
Stable pitch extraction	(Smith III & Serra, 1987), Section 5.6	Pitch distribution	`https://github.com/altugkarakurt/morty`	v1.2.1	Computed by *detect_peaks* function in *morty/pitchdistribution.py*.
Stable pitch-class extraction	(Smith III & Serra, 1987), Section 5.6	Pitch-class distribution	`https://github.com/altugkarakurt/morty`	v1.2.1	Computed by *detect_peaks* function in *morty/pitchdistribution.py*.
Tonic identification I	(Atlı et al., 2015), Section 5.7.2	Filtered predominant melody (using $\texttt{ATL-MEL}_f$)	`https://github.com/hsercanatli/tonicidentifier_makam`	v1.3.1	See `ATL-TON`. Uses the PD and PCD implementations in `MORTY`
Tonic identification II	(Karakurt et al., 2016), Section 5.7.2	Pitch-class distribution, makam (optional)	`https://github.com/altugkarakurt/morty`	v1.2.1	If makam input is not given, the task correponds to joint estimation of makam and tonic.
Transposition identification	Section 5.8	Tonic, makam or tonic Symbol	`https://github.com/sertansenturk/ahenkidentifier`	v1.5.0	
Makam recognition	(Karakurt et al., 2016), Section 5.7.2	Pitch-class distribution,tonic (optional)	`https://github.com/altugkarakurt/morty`	v1.2.1	If tonic input is not given, the task correcponds to joint estimation of makam and tonic.
Tuning analysis	(Bozkurt et al., 2009), Section 5.9	Stable pitches, makam	`https://github.com/miracatici/notemodel`	v1.2.1	Depends on `MORTY` for PD implementation and peak detection. The implementation accepts a PD, instead of the stable pitches for the sake of usability and computes the stable pitches internally. The method fails, if the scale of the makam is not available.
Melodic progression analysis	(Bozkurt, 2015), Section 5.10	Filtered predominant melody (using $\texttt{ATL-MEL}_f$)	`https://github.com/sertansenturk/seyiranalyzer`	v1.1.1	Depends on `MORTY` for PD computation
Complete audio analysis	Section 5.11	Audio file	`https://github.com/sertansenturk/tomato`	v0.9.1	Implemented in *tomato/audio*. Depends on all packages above.

Table C.3: An overview of the implementations of the joint analysis methodologies explained in Chapter 6.

Task	Methodology	Inputs	Package	Version	Comments
Fragment linking	(Şentürk, Holzapfel, & Serra, 2014), Section 6.3	Predominant melody of the audio recording (using `ATL-MEL`) using `ATL-MEL`, **SymbTr**-score fragment	`https://github.com/sertansenturk/fragmentLinker`	v0.1.0	Fundamental step in all audio-score alignment tasks/implementations. Implemented in *+fragmentLinker/@CandidateLinkEstimator/*. Synthetic melody is computed internally from the score.
Tonic identification	(Şentürk et al., 2013), Section 6.4	Predominant melody of the audio recording (using `ATL-MEL`), synthetic melody of the music score fragment	`https://github.com/sertansenturk/fragmentLinker`	v0.1.0	Implemented in *+makamLinker/TonicIdentifier*. Obtained jointly with the tempo. The binary is hosted in *tomato_binaries*, called *extractTonicTempoTuning*
Tempo estimation	(Holzapfel et al., 2015), Section 6.5	Predominant melody of the audio recording (using `ATL-MEL`), synthetic melody of the music score fragment	`https://github.com/sertansenturk/fragmentLinker`	v0.1.0	Implemented in *+makamLinker/TempoEstimator*. Obtained jointly with the tonic. The binary is hosted in *tomato_binaries*, called *extractTonicTempoTuning*
Composition identification	(Şentürk & Serra, 2016a), Section 6.6	Predominant melody of the audio recording (using `ATL-MEL`), synthetic melody of the music score fragment	`https://github.com/sertansenturk/fragmentLinker`	v0.1.0	Implemented in *+makamLinker/CompositionIdentifier*. Obtained jointly with the tonic.
Section linking	(Şentürk, Holzapfel, & Serra, 2014), Section 6.7	Predominant melody and tonic of the audio recording, synthetic melody and semiotic section labels of the music score fragment, tempo of the audio recording (optional)	`https://github.com/sertansenturk/fragmentLinker`	v0.1.0	Implemented in *+makamLinker/SectionLinker*. Obtained together with the aligned notes. The binary is hosted in *tomato_binaries*, called *alignAudioScore*. Fails if the section annotations are missing in the score.
Note-level alignment	(Şentürk, Gulati, & Serra, 2014), Section 6.8	Inputs of section linking task, section links	`https://github.com/sertansenturk/fragmentLinker`	v0.1.0	Implemented in *+makamLinker/NoteAligner*. Obtained together with the section links. The binary is hosted in *tomato_binaries*, called *alignAudioScore*
Predominant Melody Filtering	Section 6.10	Predominant melody of the audio recording (using `ATL-MEL`), aligned notes	`https://github.com/sertansenturk/alignedpitchfilter`	v1.1.0	
Note model computation	(Şentürk et al., 2016), Section 6.11	Predominant melody of the audio recording filtered with respect to the note alignments, aligned notes, Makam or Tonic Symbol	`https://github.com/sertansenturk/alignednotemodel`	v1.1.1	Depends on `MORTY` for PD implementation and peak detection.
Complete joint analysis	Section 6.12	Audio file, predominant melody of the audio recording (using `ATL-MEL`), **SymbTr** score, score features computed by *symbtrdataextractor*	`https://github.com/sertansenturk/tomato`	v1.1.1	Implemented in *tomato/joint*. Depends on all packages above.

tion is validated after each commit through automated tests using Travis CI.[31] The automated tests ease the process of adding new music scores to the collection and greatly reduce the errors accidentally introduced during score curation process. Please refer to Section 4.4.3 for the applied tests. The **SymbTr** repository also links to two other repositories as submodules:[32] The first is the *SymbTr-pdf* repository,[33] which hosts the PDFs of the music scores so that the main repository is not bulky. The second is the *SymbTr-extras* repository,[34] which contains the tools to validate and manipulate the SymbTr scores for the maintenance of the **SymbTr** repository.

In addition to test datasets, the results of most experiments conducted within the thesis are[35] are available online. Several experiments are shared step-by-step such as the makam recognition and tonic identification experiments conducted to test `MORTY` (Karakurt et al., 2016) (Section 5.7.3). Below these experiments is explained as a reference example:

The experiments in (Karakurt et al., 2016) are available online in GitHub.[36] The proper version of the test dataset is linked as a submodule.[37] All steps in the experiments (i.e. data preprocessing, training, testing and evaluation) are organized into scripts with proper documentation and instructions for reproducibility. This repository includes a setup script, which installs the appropriate version of `MORTY` and its requirements. The overall results and the evaluation are saved as json files. Due to size limitations the features, training data and testing data are hosted in Zenodo.[38] The data and the code in this repository is licensed under CC BY-NC-SA 4.0 and AGPLv3, respectively.

Finally, all the papers published within my doctoral research are

[31] `https://travis-ci.org/MTG/SymbTr`

[32] `https://git-scm.com/book/en/v2/Git-Tools-Submodules`

[33] `https://github.com/MTG/SymbTr-pdf`

[34] `https://github.com/MTG/SymbTr-extras`

[35] e.g. section linking: `http://compmusic.upf.edu/node/171`, score structure analysis: `http://compmusic.upf.edu/node/302` and makam recognition: `http://compmusic.upf.edu/node/319`.

[36] `https://github.com/sertansenturk/makam_recognition_experiments/tree/dlfm2016`

[37] see `https://github.com/sertansenturk/makam_recognition_experiments/tree/dlfm2016/data`

[38] `https://zenodo.org/record/57999`

available online via the MTG website[39] and my personal website[40] with relevant accompanying materials and documentation, when applicable.[41]

C.3 Conclusion

This Appendix gave an overview of my data, code, experiment and publication-related contributions to Dunya-makam with an aim of open research and experimental reproducibility. This Chapter is written as a "proof-of-concept" for future researchers, who would share similar concerns on open and accesible research.

Considering the rapid development in computer science, the base technologies (For example, Python 2 will abandoned in favor of Python 3 in the upcoming years) and online services, the URLs and even the digital format of this dissertation[42] will cease to exist in the upcoming years. I acknowledge that these dynamics would render some of the specific steps taken obsolete. Nevertheless, many of the concepts would continue to exist, albeit with drastic changes. It is also no doubt that the methodologies described in this thesis will be eventually replaced by faster, more powerful and more generalizable technologies in the future. I believe that the open tools, datasets and the reproducible experiments presented as part of this thesis will facilitate the proliferation and advancement of the state-of-the-art in MIR in this turn.

[39] http://mtg.upf.edu/biblio/author/494

[40] http://sertansenturk.com/research/publications/

[41] e.g. for (Karakurt et al., 2016), please visit http://mtg.upf.edu/node/3538.

[42] unless you are reading it from the sturdy, "paper" format.

Appendix D

Resources

This Appendix provides a quick access to the supplementary materials presented in the thesis. The text reflects the companion page of the thesis, hosted in the CompMusic website:

http://compmusic.upf.edu/senturk2016thesis

A mirror of the companion page is also stored in my personal website at:

http://sertansenturk.com/research/works/phd-thesis

D.1 Music Examples

These examples are compiled to show the main challenges faced in the computational analysis of OTMM such as tuning, intonation, heterophony in the performances and descriptiveness of the music scores. You have to register to Dunya-makam to listen to the audio recordings. For a thorough explanation, please refer to Chapter 2.

1. Hüseyni Peşrev by *Lavtacı Andon*: https://github.com/MTG/SymbTr-pdf/blob/v2.4.3/huseyni--pesrev--muhammes----lavtaci_andon.pdf
    - Performance by *Ahmet Kadri Rizeli*: http://dunya.compmusic.upf.edu/makam/recording/8b78115d-f7c1-4eb1-8da0-5edc564f1db3

- Performance by *İlhan Barutçu*: http://dunya.compmusic.upf.edu/makam/recording/8b78115d-f7c1-4eb1-8da0-5edc564f1db3
- Performance by *Kudsi Ergüner Ensemble*: http://dunya.compmusic.upf.edu/makam/recording/cccf944d-c237-43e0-82ac-e1c29dbc1b62

2. Muhayyer Sazsemaisi by *Tanburi Cemil Bey*: https://github.com/MTG/SymbTr-pdf/blob/v2.4.3/muhayyer--sazsemaisi--aksaksemai----tanburi_cemil_bey.pdf

   - Performance by *Necati Çelik*: http://dunya.compmusic.upf.edu/makam/recording/4948c836-5485-4a69-8bdc-11fe4559e78f
   - Performance by *Enver Mete Aslan*: http://dunya.compmusic.upf.edu/makam/recording/f51ef91d-4680-4652-8e8f-ce234e5c26e0

D.2 Research Corpus

The CompMusic Ottoman-Turkish makam music (OTMM) corpus consists of the audio recordings, music scores, metadata related to these information sources and automatic description extracted from the corpus itself. The data can be accessed from the Dunya API. The API documentation is online at:

http://dunya.compmusic.upf.edu/docs

The music scores in the corpus are taken from the **SymbTr** music score collection. The collection is maintained at:

https://github.com/MTG/SymbTr

The metadata is hosted in MusicBrainz and organized into several collections:

- **Ottoman-Turkish makam**: The releases in the CompMusic OTMM audio collection; https://musicbrainz.org/collection/5bfb724f-7e74-45fe-9beb-3e3bdb1a119e

- **Ottoman-Turkish makam excluded**: The “unrepresentative” releases in the CompMusic OTMM audio collection; https://musicbrainz.org/collection/9b7a0d92-a756-411d-81da-e855c946f23e
- **Dunya Ottoman-Turkish makam stream**: The audio recordings, which we have obtained the rights to stream in Dunya; https://musicbrainz.org/collection/af941dfc-cf39-4bbe-83f3-d367202fe629
- **SymbTr music score collection**: https://musicbrainz.org/collection/6d7ee31a-a251-4c38-b751-e0551f64c77d

D.3 Test Datasets

I have created numerous test datasets during my doctoral research:

- **OTMM Symbolic Section Dataset:** https://github.com/MTG/otmm_symbolic_section_dataset
- **OTMM Makam Recognition Dataset:** https://github.com/MTG/otmm_makam_recognition_dataset
- **OTMM Tonic Identification Datasets:** https://github.com/MTG/otmm_tonic_dataset
- **OTMM Composition Identification Dataset:** https://github.com/MTG/otmm_composition_identification_dataset
- **OTMM Section Linking Dataset:** https://github.com/MTG/otmm_section_dataset
- **OTMM Partial Audio-Score Alignment Dataset:** https://github.com/MTG/otmm_partial_alignment_dataset
- **OTMM Audio-Score Alignment Dataset:** https://github.com/MTG/otmm_audio_score_alignment_dataset

I have assisted the creation of two audio-lyrics alignment datasets:

- **OTMM Şarkı Vocal Dataset:** http://compmusic.upf.edu/node/226
- **OTMM Acapella Sections Dataset:** http://compmusic.upf.edu/turkish-makam-acapella-sections-dataset

In addition, I used several datasets created by other members of CompMusic project:

- **OTMM Symbolic Melodic Segmentation Dataset:** https://github.com/MTG/otmm_symbolic_phrase_dataset
- **Carnatic Varṇaṁ Dataset:** http://compmusic.upf.edu/carnatic-varnam-dataset
- **Carnatic Kṛti Dataset:** http://compmusic.upf.edu/node/320
- **Indian Art Music Rāga Recognition Dataset:** http://compmusic.upf.edu/node/328

D.4 Code

The implementations of the methodologies proposed to analyze the audio recordings and music scores (Chapters 4-6) are part of **T**urkish-**O**ttoman **M**akam (M)usic **A**nalysis **TO**olbox (tomato). In addition, tomato contains the tools to convert the **SymbTr** music scores to MusicXML, LilyPond and SVG formats. The toolbox is open at:

https://github.com/sertansenturk/tomato

Some complementary software tools used in the thesis are:

- Essentia audio feature extraction library: http://essentia.upf.edu/
- Dunya-web platform: https://github.com/MTG/dunya
- pycompmusic, a Python wrapper around Dunya-web API: https://github.com/MTG/pycompmusic

- Dunya-desktop, an extendible desktop interface for the navigation and annotation of music data: https://github.com/MTG/dunya-desktop

D.5 Results

The automatic description of the CompMusic OTMM corpus is open and available via the Dunya website. The data can be obtained using pycompmusic, a Python wrapper around the Dunya API.

The metadata and the automatic description are used in a web application aimed at the discovery of the CompMusic OTMM corpus. It allows the users to navigate the audio collection and play the audio recordings synchronous to the automatic description. The application is hosted in Dunya-web at:

http://dunya.compmusic.upf.edu/makam/

D.6 Publications

Please refer to Appendix E for the list of relevant publications.

D.7 Licenses

All the code presented in the thesis is licensed under GNU Affero General Public License Version 3.

All the data (e.g. the music scores, extracted features, training models, figures, text, outputs) except the copyrighted material (e.g. commercial recordings) are licensed under Creative Commons Attribution-NonCommercial-ShareAlike 4.0 License (International), unless stated otherwise.

Appendix E

List of Publications by the Author

This Appendix presents a list of my academic publications, organized by the relevance to the thesis. An up-to-date list of my publications can be found in my personal website at:

`http://sertansenturk.com/research/publications/`

- **Articles in peer-reviewed journals**

  1. Şentürk, S., Holzapfel, A., & Serra, X. (2014). Linking scores and audio recordings in makam music of Turkey. Journal of New Music Research, 43:34–52.

- **Papers published in peer-reviewed conferences**

  2. Şentürk, S., & Serra X. (2016). Composition Identification in Ottoman-Turkish Makam Music Using Transposition-Invariant Partial Audio-Score Alignment. In Proceedings of 13th Sound and Music Computing Conference (SMC 2016). pages 434–441, Hamburg, Germany
  3. Şentürk, S., Koduri G. K., & Serra X. (2016). A Score-Informed Computational Description of Svaras Using a Statistical Model. In Proceedings of 13th Sound and

Music Computing Conference (SMC 2016). pages 427-4-33, Hamburg, Germany

4. Karakurt, A., Şentürk S., & Serra X. (2016). MORTY: A Toolbox for Mode Recognition and Tonic Identification. In Proceedings of 3rd International Digital Libraries for Musicology Workshop (DLfM 2016). pages 9–16, New York, NY, USA

5. Gulati, S., Serrà J., Ganguli K. K., Şentürk S., & Serra X. (2016). Time-Delayed Melody Surfaces for Rāga Recognition. In Proceedings of 17th International Society for Music Information Retrieval Conference (ISMIR 2016), pages 751–757, New York, NY, USA

6. Şentürk S., & Serra X. (2016). A method for structural analysis of Ottoman-Turkish makam music scores. In Proceedings of 6th International Workshop on Folk Music Analysis (FMA 2016), pages 39-46, Dublin, Ireland

7. Gulati, S., Serrà J., Ishwar V., Şentürk S., & Serra X. (2016). Phrase-based Rāga Recognition Using Vector Space Modeling. In Proceedings of 41st IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP 2016), pages 66–70, Shanghai, China

8. Şentürk, S., Ferraro, A., Porter, A., & Serra, X. (2015). A tool for the analysis and discovery of Ottoman-Turkish makam music. In Extended Abstracts for the Late Breaking Demo Session of the 16th International Society for Music Information Retrieval Conference (ISMIR 2015), Málaga, Spain, 2015

9. Atlı, H. S., Bozkurt, B., Şentürk, S. (2015). A Method for Tonic Frequency Identification of Turkish Makam Music Recordings. In Proceedings of 5th International Workshop on Folk Music Analysis (FMA 2015), pages 119–122, Paris, France.

10. Holzapfel, A., Şimşekli U., Şentürk S., & Cemgil A. T. (2015). Section-level Modeling of Musical Audio for Linking Performances to Scores in Turkish Makam

Music. In Proceedings of 40th IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP 2015), pages 141–145, Brisbane, Australia.

11. Atlı, H. S., Uyar, B., Şentürk, S., Bozkurt, B., & Serra, X. (2015). Audio feature extraction for exploring Turkish makam music. In Proceedings of 3rd International Conference on Audio Technologies for Music and Media (ATMM 2015), pages 142–153, Ankara, Turkey.
12. Uyar, B., Atlı, H. S., Şentürk, S., Bozkurt, B., & Serra, X. (2014). A corpus for computational research of Turkish makam music. In Proceedings of 1st International Digital Libraries for Musicology Workshop (DLfM 2014), pages 57–63, London, United Kingdom.
13. Şentürk, S., Gulati, S., & Serra, X. (2014). Towards alignment of score and audio recordings of Ottoman-Turkish makam music. In Proceedings of 4th International Workshop on Folk Music Analysis (FMA 2014), pages 57–60, Istanbul, Turkey.
14. Şentürk, S., Gulati, S., & Serra, X. (2013). Score informed tonic identification for makam music of Turkey. In Proceedings of 14th International Society for Music Information Retrieval Conference (ISMIR 2013), pages 175–180, Curitiba, Brazil.
15. Şentürk, S., Holzapfel, A., & Serra, X. (2012). An approach for linking score and audio recordings in makam music of Turkey. In Proceedings of 2nd CompMusic Workshop, pages 95–106, Istanbul, Turkey.
16. Sordo, M., Koduri, G. K., Şentürk, S., Gulati, S., & Serra, X. (2012). A musically aware system for browsing and interacting with audio music collections. In Proceedings of 2nd CompMusic Workshop, pages 20–24, Istanbul, Turkey.

- **Publications within the CompMusic project, which are outside the context of the thesis**

15. Dzhambazov, G., Srinivasamurthy A., Şentürk S., & Serra X. (2016). On the Use of Note Onsets for Im-

proved Lyrics-to-audio Alignment in Turkish Makam Music. In Proceedings of 17th International Society for Music Information Retrieval Conference (ISMIR 2016), pages 716–722, New York, NY, USA

16. Dzhambazov, G., Şentürk, S., & Serra, X. (2015). Searching lyrical phrases in a-capella Turkish makam recordings. In Proceedings of 16th International Society for Music Information Retrieval Conference (ISMIR 2015), pages 687–693, Málaga, Spain
17. Atıcı, M., Bozkurt, B., Şentürk, S. (2015). A Culture-Specific Analysis Software for Makam Music Traditions. In Proceedings of 5th International Workshop on Folk Music Analysis (FMA 2015), pages 88–96, Paris, France.
18. Dzhambazov, G., Şentürk, S., & Serra, X. (2014). Automatic lyrics-to-audio alignment in classical Turkish music. In Proceedings of 4th International Workshop on Folk Music Analysis (FMA 2014), pages 61–64, Istanbul, Turkey.

- **Relevant publications outside the CompMusic project**

21. Chordia, P., & Şentürk, S. (2013). Joint recognition of raag and tonic in North Indian music. Computer Music Journal, 37(3):82–98.
22. Chordia, P., Sastry, A., & Şentürk, S. (2011). Predictive tabla modelling using variable-length Markov and hidden Markov models. Journal of New Music Research, 40(2):105–118.
23. Şentürk, S. (2011). Computational modeling of improvisation in Turkish folk music using variable-length Markov models. Master's thesis, Georgia Institute of Technology, Atlanta, GA, USA.
24. Şentürk, S., & Chordia, P. (2011). Modeling melodic improvisation in Turkish folk music using variable-length Markov models. In Proceedings of 12th International Society for Music Information Retrieval Conference (ISMIR 2011), pages 269–274, Miami, FL, USA.

- **Other academic publications outside the CompMusic project**

25. Şentürk, S., Lee, S. W., Sastry, A., Daruwalla, A., & Weinberg, G. (2012). Crossole: A gestural interface for composition, improvisation and performance using Kinect. In Proceedings of International Conference on New Interfaces for Musical Expression (NIME 2012), pages 449–502, Ann Arbor, MI, USA.
26. Albin, A., Şentürk, S., Van Troyer, A., Blosser, B., Jan, O., & Weinberg, G. (2011). Beatscape, a mixed virtual-physical environment for musical ensembles. In Proceedings of International Conference on New Interfaces for Musical Expression (NIME 2011), pages 112–115, Oslo, Norway.
27. Kocyigit, F. B., & Senturk, S. (2008). Acoustics in the partial deaf student school music classrooms. In Proceedings of Acoustics'08 Paris Conference, pages 3919–3924, Paris, France.
28. Şentürk, S. (2013). Sesin Özgürleşmesi: Müzik Prodüksiyonu Teknolojileri. Birikim Dergisi, 285:98–104.
29. Şentürk, S. (2011). Interactivity in Contemporary Dance and Music. Self Published.

Appendix **F**

Glossary

F.1 Technical Terms

CompMusic The research project aims to advance in the automatic description of music by emphasizing cultural specificity

CompMusic OTMM audio collection The audio collection in the CompMusic OTMM corpus

CompMusic OTMM corpus The corpus of Ottoman-Turkish makam music collected under the CompMusic project

dynamic time warping A dynamic programming algorithm to determine the similarity between two sequences. It can be back-tracked to align the sequences in time.

iterative subsequence dynamic time warping A variant of subsequence dynamic time warping, which allows multiple subsequence matching within the target for the given query.

subsequence dynamic time warping A variant of dynamic time warping, which allows subsequence matching within the target for the given query.

graph A mathematical structure depicting objects (nodes) where related objects are connected (by edges)

clique A (sub)graph with its every two node being adjacent (connected with an edge)

maximal clique A clique, which is a subclique of only itself

similar clique (Defined in the text) A maximal clique with at least one node (a structure element) different from the others (i.e. the similarity, hence edge weight is not equal to 1.

unique clique (Defined in the text) A maximal clique, where each node (a structure element) is identical with each other (i.e. the similarity, hence edge weight equals to 1)

directed acyclic graph A graph with directed edges without any directed cycles

edge A basic unit of graphs, which shows the relation between two (related) nodes

node A basic unit of graphs, which represents an "object."

subgraph A graph formed from a subset of the nodes and edges of another graph.

Hough transform A simple parametric line detection method.

link XX

F.2 Technologies

C A programming language

Dunya The music corpora and related software tools created under the CompMusic project

Dunya-desktop The interface developed by Atlı (2016)

Dunya-makam The Ottoman-Turkish makam music corpora and related software tools created under the CompMusic project

Dunya-web The web applications, which hosts and manages the corpora and the analysis tools developed under the CompMusic project

pycompmusic A wrapper around Dunya REpresentational State Transfer (REST) API

Essentia An open-source library for audio analysis and audio-based music information retrieval (Bogdanov et al., 2013)

git A version control system

LilyPond An open-source software and the relevant file format for score engraving

Makam Toolbox A makam audio analysis framework developed by Bozkurt (2011)

MakamBox Reimplementation of Makam Toolbox in Java by Atıcı et al. (2015)

MATLAB A proprietary language primarily designed for numerical computing and specifically matrix manipulations

MELODIA The predominant melody extraction method proposed by Salamon and Gómez (2012). The Essentia implementation is used throughout the thesis.

MP3 An audio coding format for digital audio, which uses a form of lossy data compression

Mus2 A notation software specialized for microtonal music and OTMM

mu2 The score file format of the score editor software, Mus2

Mus2-alfa A software by Kemal Karaosmanoğlu (Karaosmanoğlu, 2015) to notate and playback **SymbTr**-scores

MusicBrainz An open music encyclopedia of metadata

MusicXML An open format for representing digital sheet music

PostgreSQL An object-relational database

Python A high-level, interpreted programming language

Django A high-level Python Web framework

Qt An open-source application framework

Sonic Visualizer An audio analysis and visualization application developed at the Centre for Digital Music, Queen Mary, University of London

SoundFont A file format and the related technology to play MIDI files

TiMidity++ A software synthesizer for playing MIDI files

Travis CI A hosted continous integration service

Wikipedia A free online encyclopedia

F.3 Musical Concepts

heterophony Simultaneous variation of a single melodic line

mode Melodic framework. Synonymous to makam, rāg, rāga etc. in the most general sense.

scale A set of notes ordered with respect to a reference note

tonic First degree of a scale. Synonymous to the karar in OTMM or sa in IAM.

F.4 Ottoman–Turkish Makam Music

ahenk Transposition of the karar note in OTMM performances

bolahenk Default ahenk of OTMM. G4 ≈ 293 Hz

artists

Ahmet Avni Konuk (1868 / 1938) A famous composer and lyrics collector

Dede Efendi (1778 / 1846) One of the greatest composers of OTMM

Erol Bingöl (1948 /) A composer

Hacı Arif Bey (1831 / 1885) A prolific composer best known for his şarkı compositions

Rauf Yekta (1871 / 1935) A musicologist, musician and composer.

Sadettin Kaynak (1895 / 1961) A prominent composer

Şefik Gürmeriç (1904 / 1967) A composer and music theory educator

Şevki Bey (1860 / 1891) A composer, who has composed in the şarkı form

Tanburi Cemil Bey (1871 or 1873 / 1916) One of the most renowned virtuosos and composers of OTMM

başlangıç The typical melodic center in the start of a makam performance. Its literal translation is "beginning."

chord The basic building blocks of scales in OTMM

çeşni A "sample," which contains some of the peculiar characteristics of a makam (Ederer, 2011). Its literal translation is "flavor."

vuruş "Stroke" in Turkish, indicating the beats. Each vuruş has an onomatopoeic name, which describe its relative strength in an usul cycle

düm One of the onomatopoeic stroke names to refer to strong beats in an usul cycle

tek One of the onomatopoeic stroke names to refer to weak beats in an usul cycle

form The idiosyncratic structure of a music piece created by its elements

ağırsemai A classical vocal form

beste A classical, non-religious vocal form

gazel A vocal, melodic improvisation form. Typically unmetered.

ilahi The most common religious form of the classical OTMM repertoire.

küpe A classical form.

mehter The most common military form in the classical repertoire of OTMM

oyunhavası A rhythmic, instrumental folk form.

peşrev One of the most common instrumental forms of the classical OTMM repertoire.

şarkı The most common vocal form of the classical OTMM repertoire. Its literal translation is "song."

sazsemaisi One of the most common instrumental forms of the classical OTMM repertoire.

taksim An instrumental melodic improvisation form. Typically unmetered.

türkü The most common vocal form of the folk OTMM repertoire.

yürüksemai A classical, non-religious vocal form composed in yürüksemai usul.

güçlü See başlangıç

karar A melodic center and the final note where a makam performance ends. Synonymous to tonic. Its literal translatoin is "decision."

Lâedrî "Unknown" in Arabic. Used when the creator of a work (e.g. composer or lyricist of a musical work) is not known.

makam The melodic framework of Ottoman-Turkish art and folk music

Hicaz A makam with neva başlangıç and dügah karar

Hicazkar A makam with rast karar

Hüseyni A makam with hüseyni başlangıç and dügah karar

Hüzzam A makam with segah karar

Segah A makam with neva başlangıç and segah karar

Isfahan A makam with neva başlangıç and dügah karar

Kürdilihicazkar A makam with rast karar

Mahur A makam with gerdaniye başlangıç and rast karar

Muhayyer A makam similar to Hüseyni makam with "descending" melodic progression.

Neva A makam withneva başlangıç and dügah karar

Nihavent A makam with neva başlangıç and rast karar

Rast A makam with rast başlangıç and neva karar

Suzidilara A makam with rast karar

Uşşak A makam with neva başlangıç and dügah karar

mertebe The denominator of the time signature of an usul

zaman The number of pulses, i.e. the numerator of the time signature, of an usul

perde A term, which is used to refer to notes (e.g. dügah perdesi) and melodic intervals (e.g. yarım perde = semitone)

çargah The note indicated by the C5 note in the staff notation

dügah The note indicated by the A4 note in the staff notation

gerdaniye The note indicated by the G5 note in the staff notation

hüseyni The note indicated by the E5 note in the staff notation

mahur The note indicated by the G5♭ note in the staff notation

neva The note indicated by the D5 note in the staff notation

rast The note indicated by the G5 note in the staff notation

segah The note indicated by the B4𝄳 note in the staff notation

section The structural divisions, which form the compositional organization of a music piece.

aranağme An instrumental section, which is typically performed in the start of vocal forms as an introduction

hane Literal translation of section. Typically used to refer to non-repetitive sections in peşrev and sazsemaisi forms.

terennüm The repetitive vocal section in classical forms such as beste and yürüksemai. The section may include nonsensical syllables.

meyan A vocal section in the traditional OTMM composition style. The section may recapitulate the melody introduced in the zemin section.

nakarat Repetitive poetic line in şarkı and türkü forms

teslim The repetitive section in peşrev and sazsemaisi forms.

zemin First poetic line in şarkı form

seyir Melodic progression. Its literal translation is "navigation."

usul The rhythmic framework of Ottoman-Turkish art and folk music

ağıraksak An usul with 9 zaman

aksak An usul with 9 zaman and 6 vuruş

aksaksemai An usul with 10 zaman and 6 vuruş

curcuna An usul with 10 zaman

kapalı curcuna A variant of the curcuna usul with "closed" strokes

devr-i kebir An usul with 28 zaman

düyek An usul with 8 zaman and 5 vuruş

nimsofyan An usul with 2 zaman and 2 vuruş

semai An usul with 3 zaman and 3 vuruş

senginsemai Slow mertebe of the yürüksemai usul. The unit of its zaman is ♩.

serbest Non-metered. Literal translation of the word is "free."

sofyan An usul with 4 zaman and 3 vuruş

Türk aksağı An usul with 5 zaman and 3 vuruş

yürüksemai An usul with 6 zaman and 5 vuruş

F.5 Indian Art Musics

Carnatic An art music tradition of predominantly performed in south India

Hindustāni An art music tradition of predominantly performed in north India

kṛti A common vocal compositional form in Carnatic music

rāg The melodic framework of Hindustani music

rāga The melodic framework of Carnatic music

Bēgaḍa A rāga in Carnatic Music

svara The symbols used in the solfège of Indian Art Musics

sa The first position (reference svara) of Indian Art Musics

varṇaṁ A vocal compositional form in Carnatic music

F.6 Acronyms

CAR-KRI Carnatic Kṛti test dataset

CAR-VAR Carnatic Varṇaṁ test dataset

FOAF Friend of a Friend Ontology

OTMM-section-linking CompMusic Ottoman-Turkish makam music section linking test dataset

SymbTr the collection of machine-readable Ottoman-Turkish makam music scores by Karaosmanoğlu (2012)

TMKH Türk Müzik Kültürünün Hafızası (English: “Memory of Turkish Music Culture” Collection)

TRT-TTMA TRT Tarihi Türk Müziği Arşivi) (English: TRT Historical Turkish Music Archive

TSMD Türk Sanat Müziği Derlemi (English: Turkish Art Music Corpus)

UHHD Uzun Hava Humdrum Database

kNN k nearest neighbors

ATL-MEL the predominant melody extraction procedure described in (Atlı et al., 2014)

ATL-MEL$_f$ the variant of the predominant melody extraction procedure described in (Atlı et al., 2014) using the post filtering method described in (Bozkurt, 2008)

ATL-TON the tonic identification method described in (Atlı et al., 2015)

BOZ-YIN$_f$ the predominant melody extraction procedure described in (Şentürk et al., 2012), which is adapted from (Bozkurt, 2008)

CRF Conditional Random Field

DTW dynamic time warping

HHMM hierarchical hidden Markov model

ISDTW iterative subsequence dynamic time warping

LCM least common multiplier

MAP mean average precision

MORTY **MO**de **R**ecognition and **T**onic **Y**dentification Toolbox (Karakurt et al., 2016)

SDTW subsequence dynamic time warping

SEN-MEL the predominant melody extraction procedure described in (Şentürk, Holzapfel, & Serra, 2014)

SEN-MEL$_f$ the predominant melody extraction procedure described in (Şentürk, Holzapfel, & Serra, 2014) using the score-informed octave correction method described in Section 6.10

SEN-TON$_{PCD}$ the tonic identification method by score template distribution matching (Şentürk et al., 2013)

SEN-TON$_{link}$ the tonic identification method by fragment linking (Şentürk et al., 2013)

VLMM variable-length Markov model

VMD variational mode decomposition

`YIN` the fundamental pitch extraction method proposed by (De Cheveigné & Kawahara, 2002)

`tomato` **T**urkish-**O**ttoman **M**akam (M)usic **A**nalysis **TO**olbox

`SIM-VMD` the predominant melody extraction method proposed by (Şimşek et al., 2016)

AEU theory Arel-Ezgi-Uzdilek theory

AGPLv3 GNU Affero General Public License Version 3

API Application Programming Interface

bpm beat per minute

CC BY-NC 3.0 Creative Commons Attribution-NonCommercial 3.0 License

CC BY-NC-SA 4.0 Creative Commons Attribution-NonCommercial-ShareAlike 4.0 License

CQT Constant Q transform

DAG directed acyclic graph

DOI digital object identifier

Hc Holderian comma

HMM Hidden Markov model

HPCP harmonic pitch class profile (Gómez, 2006)

Hz Hertz

IAM Indian Art Musics

JSON JavaScript object notation

kbps kilobits per second

MBID MusicBrainz identifier

MIDI musical instrument digital interface

MIR music information research

MIREX Music Information Retrieval EXchange

OTMM Ottoman-Turkish makam music

PCD pitch-class distribution

PD pitch distribution

PDF portable document format

REST REpresentational State Transfer

SVG scalable vector graphics format

TET tone-equal-tempered

TRT Türkiye Radyo ve Televizyon Kurumu (English: Turkish Radio and Television Corporation)

TSV tab separated values

URL uniform resource locator

XML extensible markup language

Bibliography

Abdoli, S. (2011). Iranian traditional music Dastgah classification. In *Proceedings of the 12th International Society for Music Information Retrieval Conference (ISMIR 2011)* (pp. 275–280). Miami, FL, USA. [95]

Abesser, J., Frieler, K., Cano, E., Pfleiderer, M., & Zaddach, W. G. (2016). Score-informed analysis of tuning, intonation, pitch modulation, and dynamics in jazz solos. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, *PP*(99). [14, 123, 203]

Anguera, X., & Ferrarons, M. (2013). Memory efficient subsequence DTW for Query-by-Example spoken term detection. In *Proceedings of IEEE International Conference on Multimedia and Expo (ICME 2013)* (pp. 1–6). San Jose, CA, USA. [135, 240]

Arel, H. S. (1968). *Türk musikisi nazariyatı (the theory of turkish music)*. ITMKD yayınları. [8]

Arthur, D., & Vassilvitskii, S. (2007). k-means++: The advantages of careful seeding. In *Proceedings of the 18th Annual ACM-SIAM Symposium on Discrete Algorithms* (pp. 1027–1035). New Orleans, LA, USA. [199]

Arzt, A., Böck, S., & Widmer, G. (2012). Fast identification of piece and score position via symbolic fingerprinting. In *Proceedings of 13th International Society for Music Information Retrieval Conference (ISMIR 2012)* (pp. 433–438). Porto, Portugal. [14, 16, 123]

Arzt, A., & Widmer, G. (2010). Towards effective ‘any-time’ music tracking. In *Proceedings of the Starting AI Researchers’ Symposium (STAIRS 2010)* (pp. 24–36). Lisbon, Portugal. [16]

Arzt, A., Widmer, G., & Sonnleitner, R. (2014). Tempo- and transposition-invariant identification of piece and score position. In *Proceedings of the international society for music information retrieval conference (ismir).* [14, 16, 203]

Atalay, N. B., & Yöre, S. (2011). *Türk sanat müziği derlemi.* Retrieved from `www.tsmderlemi.com` [31]

Atıcı, B. M., Bozkurt, B., & Şentürk, S. (2015). A culture-specific analysis software for makam music traditions. In *Proceedings of 5th International Workshop on Folk Music Analysis (FMA 2015)* (pp. 88–92). Paris, France. [80, 277]

Atlı, H. S. (2016). *Türk makam müziği’nin ezgisel boyutuna yönelik İnteraktif eğitim programı* (Unpublished master’s thesis). Bahçeşehir Üniversitesi. [xli, xlvi, 46, 88, 89, 212, 213, 276]

Atlı, H. S., Bozkurt, B., & Şentürk, S. (2015). A method for tonic frequency identification of Turkish makam music recordings. In *Proceedings of 5th International Workshop on Folk Music Analysis (FMA 2015)* (pp. 119–122). Paris, France. [xli, 44, 91, 96, 97, 105, 107, 108, 110, 112, 121, 259, 282]

Atlı, H. S., Uyar, B., Şentürk, S., Bozkurt, B., & Serra, X. (2014). Audio feature extraction for exploring Turkish makam music. In *Proceedings of 3rd International Conference on Audio Technologies for Music and Media (ATMM 2014)* (p. 142--153). Ankara, Turkey. [xli, 44, 46, 86, 88, 105, 108, 113, 259, 282]

Aucouturier, J.-J., & Sandler, M. (2002). Finding repeating patterns in acoustic musical signals: Applications for audio thumbnailing. In *Proceedings of 22nd International Audio Engineering Society Conference: Virtual, Synthetic, and Entertainment Audio.* Espoo, Finland. [132, 185]

Ayangil, R. (2001). 21. yüzyıl eşiğinde Türkiye’de müzik kuramı çalışmaları. *Musikişinas Dergisi*(5), 72–81. [10]

Ayangil, R. (2008). Western notation in Turkish music. *Journal of the Royal Asiatic Society (Third Series)*, *18*(04), 401–447. [9]

Ballard, D. H. (1981). Generalizing the Hough transform to detect arbitrary shapes. *Pattern recognition*, *13*(2), 111–122. [132, 219, 220, 231, 239]

Başaran, D., Cemgil, A. T., & Anarım, E. (2015). A probabilistic model-based approach for aligning multiple audio sequences. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, *23*(7), 1160–1171. [203]

Behar, C. (2015). *Osmanlı - türk musıkisinin kısa tarihi*. Yapı Kredi Yayınları. [7, 8, 11]

Benetos, E., & Holzapfel, A. (2013). Automatic transcription of Turkish makam music. In *Proceedings of 12th International Conference on Music Information Retrieval (ISMIR 2013)* (pp. 355–360). Curitiba, Brazil. [188]

Benetos, E., & Holzapfel, A. (2015). Automatic transcription of Turkish microtonal music. *Journal of the Acoustical Society of America*, *138*(4), 2118-2130. [14, 46, 95, 121]

Bertin-Mahieux, T., Ellis, D. P., Whitman, B., & Lamere, P. (2011). The million song dataset. In *Proceedings of the 12th International Conference on Music Information Retrieval (ISMIR 2011)* (pp. 591–596). Miami, FL, USA. [13]

Bimbot, F., Deruty, E., Sargent, G., & Vincent, E. (2012). Semiotic structure labeling of music pieces: Concepts, methods and annotation conventions. In *Proceedings of the 13th International Society for Music Information Retrieval Conference (ISMIR 2012)* (pp. 235–240). Porto, Portugal. [66]

Böck, S., Korzeniowski, F., Schlüter, J., Krebs, F., & Widmer, G. (2016). Madmom: A new python audio and music signal processing library. In *Proceedings of the 2016 ACM on Multimedia Conference* (pp. 1174–1178). New York, NY, USA. [79]

Bod, R. (2002). Memory-based models of melodic analysis: Challenging the gestalt principles. *Journal of New Music Research*, *31*(1), 27–36. [59]

Bogdanov, D., Wack, N., Gómez, E., Gulati, S., Herrera, P., Mayor, O., Roma, G., Salamon, J., Zapata, J., & Serra, X. (2013). Essentia: An audio analysis library for music information retrieval. In *Proceedings of 14th International Society for Music Information Retrieval Conference (ISMIR 2013)* (pp. 493–498). Curitiba, Brazil. [58, 79, 85, 253, 255, 276]

Bozkurt, B. (2008). An automatic pitch analysis method for Turkish maqam music. *Journal of New Music Research*, *37*(1), 1–13. [80, 83, 84, 88, 91, 93, 96, 98, 100, 107, 110, 112, 113, 178, 216, 253, 259, 282]

Bozkurt, B. (2011). Pitch histogram based analysis of makam music in Turkey. In *Proceedings of les corpus de l' oralité.* Strasbourg, France. [79, 276]

Bozkurt, B. (2012). A system for tuning instruments using recorded music instead of theory-based frequency presets. *Computer Music Journal*, *36*, 43–56. [80, 95]

Bozkurt, B. (2015). Computational analysis of overall melodic progression for Turkish makam music. In *Penser l'improvisation* (p. 266-290). Sampzon, France: Delatour France. [10, 116, 117, 259]

Bozkurt, B., Ayangil, R., & Holzapfel, A. (2014). Computational analysis of Turkish makam music: Review of state-of-the-art and challenges. *Journal of New Music Research*, *43*(1), 3-23. [7, 10, 11, 13, 42, 81, 83, 120]

Bozkurt, B., Karaosmanoğlu, M. K., Karaçalı, B., & Ünal, E. (2014). Usul and makam driven automatic melodic segmentation for Turkish music. *Journal of New Music Research*, *43*(4), 375-389. [42, 49, 59, 62, 68, 69, 72, 191, 256, 258]

Bozkurt, B., Yarman, O., Karaosmanoğlu, M. K., & Akkoç, C. (2009). Weighing diverse theoretical models on Turkish maqam music against pitch measurements: A comparison of peaks automatically derived from frequency histograms with proposed scale tones. *Journal of New Music Research*, *38*(1), 45–70. [9, 91, 95, 114, 115, 116, 160, 203, 259]

Brickley, D., & Miller, L. (2014). *FOAF vocabulary specification.* Retrieved from http://xmlns.com/foaf/spec/ [47]

Bühlmann, P., & Wyner, A. J. (1999). Variable length Markov chains. *The Annals of Statistics*, *27*(2), 480–513. [166]

Cambouropoulos, E. (2001). The local boundary detection model (LBDM) and its application in the study of expressive timing. In *Proceedings of the International Computer Music Conference (ICMC 2001)* (pp. 17–22). La Habana, Cuba. [58, 59]

Casey, M. A., Veltkamp, R., Goto, M., Leman, M., Rhodes, C., & Slaney, M. (2008). Content-based music information retrieval: Current directions and future challenges. *Proceedings of the IEEE*, *96*(4), 668-696. [2]

Cha, S.-H., & Srihari, S. N. (2002). On measuring the distance between histograms. *Pattern Recognition*, *35*(6), 1355–1370. [101]

Chandola, V., Banerjee, A., & Kumar, V. (2009). Anomaly detection: A survey. *ACM Computing Surveys*, *41*(3), 15:1–15:58. [152]

Chordia, P., & Şentürk, S. (2013). Joint recognition of raag and tonic in North Indian music. *Computer Music Journal*, *37*(3). [xxxv, 83, 92, 93, 95, 96, 97, 98, 99, 100, 101, 106, 111, 113, 121, 138, 140, 195, 233, 236, 243, 248, 249, 250, 259]

Chordia, P., & Rae, A. (2007). Raag recognition using pitch-class and pitch-class dyad distributions. In *Proceedings of 8th International Conference on Music Information Retrieval (ISMIR 2007)* (pp. 431–436). Vienna, Austria. [95]

Cont, A. (2010). A coupled duration-focused architecture for realtime music-to-score alignment. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *32*(6), 974–987. [14, 123, 238]

Cooke, P. (accessed April 5, 2013). *Heterophony.* Grove Music Online. http://www.oxfordmusiconline.com/subscriber/article/grove/music/12945. Oxford University Press. [8]

Cooper, M., & Foote, J. (2002). Automatic music summarization via similarity analysis. In *Proceedings of 3rd International Society for Music Information Retrieval Conference (ISMIR 2002)* (pp. 81–85). Paris, France. [15]

Cornelis, O., Lesaffre, M., Moelants, D., & Leman, M. (2010). Access to ethnic music: Advances and perspectives in content-based music information retrieval. *Signal Processing*, *90*(4), 1008–1031. [186]

Davies, M. E., & Plumbley, M. D. (2007). Context-dependent beat tracking of musical audio. *IEEE Transactions on Audio, Speech, and Language Processing*, *15*(3), 1009–1020. [250]

De Cheveigné, A., & Kawahara, H. (2002). YIN, a fundamental frequency estimator for speech and music. *Journal of Acoustical Society of America*, *111*(4), 1917–1930. [83, 283]

Devaney, J., Mandel, M., & Fujinaga, I. (2012). A study of intonation in three-part singing using the automatic music performance analysis and comparison toolkit (AMPACT). In *Proceedings of 13th International Society for Music Information Retrieval Conference (ISMIR 2012)* (pp. 511–516). Porto, Portugal. [14, 15, 16, 123]

Dighe, P., Karnick, H., & Raj, B. (2013). Swara histogram based structural analysis and identification of indian classical ragas. In *Proceedings of the 14th International Society for Music Information Retrieval Conference (ISMIR 2013)* (pp. 35–40). Curitiba, Brazil: Pontifícia Universidade Católica do Paraná. [95]

Dixon, S., & Widmer, G. (2005). Match: A music alignment tool chest. In *Proceedings of 6th International Society for Music Information Retrieval Conference (ISMIR 2005)* (pp. 492–497). London, United Kingdom. [14, 238]

Dorfer, M., Arzt, A., & Widmer, G. (2016). Towards score following in sheet music images. In *Proceedings of the International Society for Music Information Retrieval Conference (ISMIR 2016)* (pp. 789–794). New York, NY, USA. [14]

Duan, Z., & Pardo, B. (2011). Aligning semi-improvised music audio with its lead sheet. In *Proceedings of 12th International Society for Music Information Retrieval Conference (ISMIR 2011)* (pp. 513–518). Miami, FL, USA. [16]

Duda, R. O., & Hart, P. E. (1972). Use of the Hough transformation to detect lines and curves in pictures. *Communications of the ACM*, *15*(1), 11–15. [133, 160]

Dzhambazov, G., Şentürk, S., & Serra, X. (2014). Automatic lyrics-to-audio alignment in classical Turkish music. In *Proceedings of 4th International Workshop on Folk Music Analysis (FMA 2014)* (pp. 61–64). Istanbul, Turkey. [46, 49, 212]

Dzhambazov, G., & Serra, X. (2015). Modeling of phoneme durations for alignment between polyphonic audio and lyrics. In *Proceedings of Sound and Music Computing Conference 2015 (SMC 2015)*. Maynooth, Ireland. [46, 212]

Dzhambazov, G., Srinivasamurthy, A., Şentürk, S., & Serra, X. (2016). On the use of note onsets for improved lyrics-to-audio alignment in Turkish makam music. In *Proceedings of 17th International Society for Music Information Retrieval Conference (ISMIR 2016)* (pp. 716–722). New York, NY, USA. [46, 77, 192, 212]

Ederer, E. B. (2011). *The theory and praxis of makam in classical Turkish music 1910-2010* (Unpublished doctoral dissertation). University of California, Santa Barbara. [8, 9, 10, 278]

Ellis, D. P., & Poliner, G. E. (2007). Identifying 'cover songs' with chroma features and dynamic programming beat tracking. In *Proceedings of the 32nd IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP 2007)* (pp. 1429–1432). Honolulu, HI, USA. [2, 17, 135, 148]

Ewert, S., & Müller, M. (2012). Score-informed source separation for music signals. In *Multimodal Music Processing* (Vol. 3, pp. 73–94). Dagstuhl, Germany: Schloss Dagstuhl–Leibniz-Zentrum für Informatik. [123]

Fremerey, C., Müller, M., & Clausen, M. (2010). Handling repeats and jumps in score-performance synchronization. In *Proceedings of 11th International Society for Music Information Retrieval Conference (ISMIR 2010)* (pp. 243–248). Utrecht, Netherlands. [14, 15, 16, 186, 191, 238, 239]

Fujihara, H., & Goto, M. (2012). Lyrics-to-audio alignment and its application. In *Multimodal music processing* (Vol. 3, pp. 23–36). Dagstuhl, Germany: Schloss Dagstuhl–Leibniz-Zentrum für Informatik. [2]

Fujishima, T. (1999). Realtime chord recognition of musical sound: A system using Common Lisp Music. In *Proceedings of International Computer Music Conference (ICMC 1999)* (pp. 464–467). Beijing, China. [195]

Gedik, A. C. (2012). *Automatic transcription of traditional Turkish art music recordings: A computational ethnomusicology approach* (Unpublished doctoral dissertation). Izmir Institute of Technology, Izmir, Turkey. [80]

Gedik, A. C., & Bozkurt, B. (2010). Pitch-frequency histogram-based music information retrieval for Turkish music. *Signal Processing*, *90*(4), 1049–1063. [xxxv, 80, 91, 92, 95, 96, 97, 98, 100, 101, 102, 106, 107, 108, 111, 113, 121, 138, 140, 144, 145, 160, 195, 216]

Gómez, E. (2006). *Tonal description of music audio signals* (Unpublished doctoral dissertation). Universitat Pompeu Fabra.

[xxxi, xxxii, 58, 82, 89, 91, 195, 283]

Goto, M. (2003). A chorus-section detecting method for musical audio signals. In *Proceedings of the 28th IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP 2003)* (Vol. 5, pp. 437–40). Hong Kong, Hong Kong. [15]

Goto, M., Hashiguchi, H., Nishimura, T., & Oka, R. (2003). Rwc music database: Music genre database and musical instrument sound database. In *Proceedings of 4th International Society for Music Information Retrieval Conference (ISMIR 2003)* (pp. 229–230). Baltimore, MD, USA. [13]

Grachten, M., Gasser, M., Arzt, A., & Widmer, G. (2013). Automatic alignment of music performances with structural differences. In *Proceedings of 14th International Society for Music Information Retrieval Conference (ISMIR 2013)* (pp. 607–612). Curitiba, Brazil. [16, 191]

Gruber, T. R. (1995). Toward principles for the design of ontologies used for knowledge sharing. *International Journal of Human-Computer Studies*, *43*(5), 907–928. [47]

Gulati, S. (2011). *A Tonic Identification Approach for Indian Art Music* (Master's Thesis). Universitat Pompeu Fabra, Barcelona, Spain. [96, 240, 248]

Gulati, S. (2016). *Computational approaches for melodic description in indian art music corpora* (Unpublished doctoral dissertation). Universitat Pompeu Fabra, Barcelona. [248]

Gulati, S., Serrà, J., Ganguli, K. K., Şentürk, S., & Serra, X. (2016). Time-delayed melody surfaces for rāga recognition. In *Proceedings of 17th International Society for Music Information Retrieval Conference (ISMIR 2016)* (pp. 751–757). New York, NY, USA. [xlix, 95, 104, 114, 122, 248, 249, 250]

Gulati, S., Serrà, J., Ishwar, V., Şentürk, S., & Serra, X. (2016). Phrase-based rāga recognition using vector space modeling. In *Proceedings of 41st IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP 2016)* (p. 66-70). Shanghai, China: IEEE. [xlix, 95, 104, 112, 248, 249, 250]

Hagberg, A. A., Schult, D. A., & Swart, P. J. (2008). Exploring network structure, dynamics, and function using NetworkX. In *Proceedings of the 7th Python in Science Confer-*

ence (SciPy2008) (pp. 11–15). Pasadena, CA, USA. [255]

Holzapfel, A. (2010). *Similarity methods for computational ethnomusicology* (Unpublished doctoral dissertation). University of Crete. [2]

Holzapfel, A. (2015a). Melodic key phrases in traditional cretan dance tunes. In *Proceedings of the 5th Workshop on Folk Music Analysis (FMA 2015)* (p. 79-82). Paris, France. [250, 251, 252]

Holzapfel, A. (2015b). Relation between surface rhythm and rhythmic modes in turkish makam music. *Journal for New Music Research*, *44*(1), 25-38. [203]

Holzapfel, A. (in press). Rhythmic and melodic aspects of Cretan leaping dances. In *Music on Crete, Traditions of a Mediterranean Island.* Vienna Series in Ethnomusicology. [xlvi, 250, 251]

Holzapfel, A., Davies, M. E. P., Zapata, J. R., Oliveira, J. L., & Gouyon, F. (2012). On the automatic identification of difficult examples for beat tracking: Towards building new evaluation datasets. In *Proceedings of the 37th IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP 2012)* (pp. 89–92). Kyoto, Japan. [44]

Holzapfel, A., Şimşekli, U., Şentürk, S., & Cemgil, A. T. (2015). Section-level modeling of musical audio for linking performances to scores in Turkish makam music. In *Proceedings of the 40th IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP 2015)* (pp. 141–145). Brisbane, Australia. [14, 16, 45, 147, 180, 181, 203, 239, 248, 260]

Holzapfel, A., & Stylianou, Y. (2009). Rhythmic similarity in traditional Turkish music. In *Proceedings of 10th International Society for Music Information Retrieval Conference (ISMIR 2009)* (pp. 99–104). Kobe, Japan. [17]

Hunter, J. D. (2007). Matplotlib: A 2D graphics environment. *Computing in Science & Engineering*, *9*(3), 90-95. [255]

Jackendoff, R. (1985). *A generative theory of tonal music*. MIT Press. [58]

Joder, C., Essid, S., & Member, S. (2010). A conditional random field framework for robust and scalable audio-to-score matching. *IEEE Transactions on Audio, Speech and Lan-*

guage Processing, *19*(8), 2385–2397. [14, 16]

Jones, E., Oliphant, T., Peterson, P., et al. (2001–). *SciPy: Open source scientific tools for Python.* Retrieved from http://www.scipy.org/ [255]

Karadeniz, M. E. (1984). Türk musıkisinin nazariye ve esasları. In *Türk musikisinde kullanılmış olan beste çeşitleri* (p. 159). İş Bankası Yayınları (in Turkish). [8, 11]

Karakurt, A., Şentürk, S., & Serra, X. (2016). MORTY: A toolbox for mode recognition and tonic identification. In *Proceedings of the 3rd International Digital Libraries for Musicology Workshop (DLfM 2016)* (pp. 9–16). New York, NY, USA. [43, 44, 97, 98, 104, 138, 140, 248, 254, 259, 261, 262, 282]

Karaosmanoğlu, M. K. (2012). A Turkish makam music symbolic database for music information retrieval: SymbTr. In *Proceedings of 13th International Society for Music Information Retrieval Conference (ISMIR 2012)* (pp. 223–228). Porto, Portugal. [xxxix, 26, 27, 28, 37, 58, 154, 172, 188, 216, 224, 225, 281]

Karaosmanoğlu, M. K., Bozkurt, B., Holzapfel, A., & Doğrusöz Dişiaçık, N. (2014). A symbolic dataset of Turkish makam music phrases. In *Proceedings of 4th International Workshop on Folk Music Analysis (FMA 2014)* (pp. 10–14). Istanbul, Turkey. [42, 43, 62, 63]

Karaosmanoğlu, M. K. (2015). *Türk musikisi semboli verileri Üzerinde hesaplamalı ezgi analizi* (Unpublished doctoral dissertation). Yıldız Teknik Üniversitesi. [27, 28, 37, 38, 277]

Kimmel, R., & Sethian, J. A. (1998). Computing geodesic paths on manifolds. *Proceedings of the National Academy of Sciences*, *95*(15), 8431-8435. [231]

Klir, G. J., & Yuan, B. (1995). *Fuzzy sets and fuzzy logic: Theory and applications*. Prentice Hall New Jersey. [231]

Koduri, G. K., Gulati, S., Rao, P., & Serra, X. (2012). Rāga recognition based on pitch distribution methods. *Journal of New Music Research*, *41*(4), 337–350. [95, 236, 240]

Koduri, G. K., Ishwar, V., Serrà, J., & Serra, X. (2014). Intonation analysis of rāgas in Carnatic music. *Journal of New Music Research*, *43*(01), 72–93. [83, 95, 195, 234, 235, 236, 240, 243, 245, 246, 247, 248, 249]

Krishna, T. M., & Ishwar, V. (2012). Karṇāṭik music: Svara, gamaka, phraseology and rāga identity. In *Proceedings of 2nd CompMusic Workshop* (pp. 12–18). Istanbul, Turkey. [234]

Krishnaswamy, A. (2003). On the twelve basic intervals in south Indian classical music. *Audio Engineering Society Convention*, 1–14. [236, 237]

Kroher, N., Díaz-Báñez, J.-M., Mora, J., & Gómez, E. (2016). Corpus COFLA: A Research Corpus for the Computational Study of Flamenco Music. *Journal on Computing and Cultural Heritage (JOCCH)*, *9*(2), 10:1–10:21. Retrieved from http://doi.acm.org/10.1145/2875428 [13]

Krumhansl, C. L., & Shepard, R. N. (1979). Quantification of the hierarchy of tonal functions within a diatonic context. *Journal of Experimental Psychology: Human Perception and Performance*, *5*(4), 579–594. [91]

Lartillot, O., & Ayari, M. (2009). Segmentation of Tunisian modal improvisation: Comparing listeners' responses with computational predictions. *Journal of New Music Research*, *38*(2), 117–127. [58]

Lartillot, O., Toiviainen, P., & Eerola, T. (2008). A Matlab toolbox for music information retrieval. In *Data analysis, machine learning and applications* (pp. 261–268). Springer. [79]

Lartillot, O., Yazıcı, Z. F., & Mungan, E. (2013). A pattern-expectation, non-flattening accentuation model, empirically compared with segmentation models on traditional Turkish music. In *Proceedings of the 3rd International Workshop on Folk Music Analysis (FMA 2013)* (pp. 63–70). Amsterdam, Netherlands. [59]

League, P. (2012). Çeçen Kızı: Tracing a Tune through the Ottoman Ecumene. *Portfolio of the Department of Musicology and Ethnomusicology*, *1*. [27]

Levenshtein, V. I. (1966). Binary codes capable of correcting deletions, insertions, and reversals. *Soviet Physics Doklady*, *10*(8), 707–710. [64]

Lin, J., Keogh, E., Lonardi, S., & Chiu, B. (2003). A symbolic representation of time series, with implications for streaming algorithms. In *Proceedings of the 8th ACM SIGMOD Workshop on Research Issues in Data Mining and Knowl-*

edge Discovery (DMKD'03) (pp. 2–11). New York, NY, USA. [231]

MacKay, D. J. C. (2003). *Information theory, inference and learning algorithms*. Cambridge University Press. [242]

Maesschalck, R. D., Jouan-Rimbaud, D., & Massart, D. (2000). The mahalanobis distance. *Chemometrics and Intelligent Laboratory Systems*, *50*(1), 1 - 18. [153]

Maezawa, A., Itoyama, K., Yoshii, K., & Okuno, H. G. (2014). Bayesian audio alignment based on a unified generative model of music composition and performance. In *Proceedings of the 15th International Society for Music Information Retrieval Conference (ISMIR 2014)* (pp. 233–238). Taipei, Taiwan. [14]

Maezawa, A., Okuno, H. G., Ogata, T., & Goto, M. (2011). Polyphonic audio-to-score alignment based on Bayesian latent harmonic allocation hidden Markov model. In *Proceedings of the 36th IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP 2011)* (pp. 185–188). Prague, Czech Republic. [238]

Manning, C. D., Raghavan, P., & Schütze, H. (2008). *Introduction to information retrieval* (Vol. 1). Cambridge University Press. [152, 153, 154]

Marcus, S. (2001). Rhythmic modes in middle eastern music. In *Garland encyclopedia of world music* (pp. 89–92). Alexander Street. [10]

Martin, B., Robine, M., & Hanna, P. (2009). Musical structure retrieval by aligning self-similarity matrices. In *Proceedings of the 10th International Society for Music Information Retrieval Conference (ISMIR 2009)* (pp. 483–488). Utrecht, Netherlands. [2]

McFee, B., McVicar, M., Raffel, C., Liang, D., Nieto, O., Battenberg, E., Moore, J., Ellis, D., Yamamoto, R., Bittner, R., Repetto, D., Viktorin, P., Santos, J. F., & Holovaty, A. (2015). *librosa: 0.4.1.* Retrieved from https://doi.org/10.5281/zenodo.32193 [79]

McKay, C. (2010). *Automatic music classification with jMIR* (Unpublished doctoral dissertation). McGill University. [79]

McKinney, W. (2010). Data structures for statistical computing in Python. In *Proceedings of the 9th Python in Science Confer-*

ence (SciPy2010) (pp. 51–56). Austin, TX, USA. [255]

Moelants, D., Cornelis, O., Leman, M., Gansemans, J., De Caluwe, R., De Tré, G., Matthé, T., & Hallez, A. (2007). The problems and opportunities of content-based analysis and description of ethnic music. *International Journal of Intangible Heritage*, *2*, 57–68. [12]

Müller, M. (2007). *Information retrieval for music and motion* (Vol. 6). Springer Heidelberg. [135, 187, 241]

Müller, M., & Appelt, D. (2008). Path-constrained partial music synchronization. In *Proceedings of the 33rd IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP 2008)* (pp. 65–68). Las Vegas, NV, USA. [16, 135, 240]

Müller, M., & Clausen, M. (2007). Transposition-invariant self-similarity matrices. In *Proceedings of the 8th International Conference on Music Information Retrieval (ISMIR 2007)* (pp. 47–50). Vienna, Austria. [17]

Müller, M., & Ewert, S. (2008). Joint structure analysis with applications to music annotation and synchronization. In *Proceedings of the 9th International Conference on Music Information Retrieval (ISMIR 2008)* (pp. 389–394). Philadelphia, PA, USA. [2]

Müller, M., Ewert, S., & Kreuzer, S. (2009). Making chroma features more robust to timbre changes. In *Proceedings of the 34th IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP 2009)* (pp. 1877–1880). Taipei, Taiwan. [82, 229]

Müller, M., Grosche, P., & Wiering, F. (2009). Towards automated processing of folk song recordings. In *Knowledge representation for intelligent music processing.* Dagstuhl, Germany: Schloss Dagstuhl - Leibniz-Zentrum fuer Informatik, Germany. [16, 17]

Nakamura, T., Nakamura, E., & Sagayama, S. (2016). Real-time audio-to-score alignment of music performances containing errors and arbitrary repeats and skips. *IEEE/ACM Trans. Audio, Speech and Lang. Proc.*, *24*(2), 329–339. [14, 16]

Newman, M. E. J. (2010). *Networks: An Introduction*. Oxford University Press. [17]

Niedermayer, B. (2009). Towards audio to score alignment in the symbolic domain. In *Proceedings of the sound and music computing conference (smc)* (pp. 77–82). [16]

Niedermayer, B. (2012). *Accurate Audio-to-Score Alignment – Data Acquisition in the Context of Computational Musicology* (Unpublished doctoral dissertation). Johannes Kepler Universität. [2, 14, 135, 238]

Nienhuys, H.-W., & Nieuwenhuizen, J. (2003). LilyPond, a system for automated music engraving. In *Proceedings of the XIV Colloquium on Musical Informatics (XIV CIM 2003)* (Vol. 1, pp. 167–171). Firenze, Italy. [255]

Özkan, I. H. (2006). *Türk mûsikısi nazariyatı ve usûlleri: Kudüm velveleleri*. Ötüken Neşriyat (in Turkish). [8, 113, 115]

Pardo, B., & Birmingham, W. (2005). Modeling form for on-line following of musical performances. In *Proceedings of the 20th national conference on artificial intelligence - volume 2* (pp. 1018–1023). AAAI Press. [16]

Paulus, J., & Klapuri, A. (2009). Music structure analysis using a probabilistic fitness measure and a greedy search algorithm. *IEEE Transactions on Audio, Speech, and Language Processing*, *17*(6), 1159–1170. [17]

Paulus, J., Müller, M., & Klapuri, A. (2010). State of the art report: Audio-based music structure analysis. In *Proceedings of 11th International Society for Music Information Retrieval Conference (ISMIR 2010)* (pp. 625–636). Utrecht, Netherlands. [14, 15, 82, 131]

Pearce, M. T., Müllensiefen, D., & Wiggins, G. A. (2010). Melodic grouping in music information retrieval: New methods and applications. In *Advances in music information retrieval* (pp. 364–388). Springer. [58, 59]

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., & Duchesnay, E. (2011). Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, *12*(Oct), 2825–2830. [255]

Peeling, P. H., Cemgil, A. T., & Godsill, S. J. (2007). A probabilistic framework for matching music representations. In *Ismir.* Vienna, Austria. [14]

Pikrakis, A., Theodoridis, S., & Kamarotos, D. (2003). Recognition of isolated musical patterns using context dependent dynamic time warping. *IEEE Transactions on Speech and Audio Processing*, *11*(3), 175–183. [2]

Popescu-Judetz, E. (1996). *Meanings in Turkish musical culture*. Istanbul: Pan Yayıncılık. [9]

Porter, A., Bogdanov, D., Kaye, R., Tsukanov, R., & Serra, X. (2015). AcousticBrainz: a community platform for gathering music information obtained from audio. In *16th International Society for Music Information Retrieval Conference (ISMIR 2015)* (pp. 786–792). Malaga, Spain. [13]

Porter, A., & Serra, X. (2014). An analysis and storage system for music research datasets. In *Proceedings of 1st International Digital Libraries for Musicology Workshop (DLfM 2014)* (p. 1-3). London, United Kingdom. [208]

Porter, A., Sordo, M., & Serra, X. (2013a). Dunya: A system for browsing audio music collections exploiting cultural context. In *Proceedings of 14th International Society for Music Information Retrieval Conference (ISMIR 2013).* Curitiba, Brazil. [25, 26, 186]

Porter, A., Sordo, M., & Serra, X. (2013b). Dunya: A system for browsing audio music collections exploiting cultural context. In *Proceedings of 14th International Society for Music Information Retrieval Conference (ISMIR 2013)* (p. 101-106). Curitiba, Brazil. [208]

Powers, et al., H. S. (n.d.). *Mode.* Grove Music Online, Oxford Music. Online: `http://www.oxfordmusiconline.com/subscriber/article/grove/music/43718pg5S`. [95]

Prätzlich, T., Driedger, J., & Müller, M. (2016). Memory-restricted multiscale dynamic time warping. In *Proceedings of the IEEE international conference on acoustics, speech, and signal processing (ICASSP)* (pp. 569–573). Shanghai, China. [14]

Prätzlich, T., & Müller, M. (2014). Frame-level audio segmentation for abridged musical works. In *Proceedings of the 15th International Society for Music Information Retrieval Conference (ISMIR 2014)* (pp. 307–312). Taipei, Taiwan. [16]

Pérez, F., & Granger, B. E. (2007). IPython: A system for interactive scientific computing. *Computing in Science & Engineering*, *9*(3), 21-29. [256]

Rafael, R. C., & Serra, X. (2014). Creating a corpus of jingju (beijing opera) music and possibilities for melodic analysis. In *15th international society for music information retrieval conference* (p. 313-318). Taipei, Taiwan. [13]

Raffel, C. (2016). *Learning-based methods for comparing sequences, with applications to audio-to-MIDI alignment and matching* (Unpublished doctoral dissertation). Columbia University. [203]

Raimond, Y. (2008). *A Distributed Music Information System* (Unpublished doctoral dissertation). Queen Mary, University of London. [47]

Raimond, Y., Abdallah, S., Sandler, M., & Frederick, G. (2007). The Music Ontology. In *Proceedings of the 6th International Conference on Music Information Retrieval (ISMIR 2007)* (pp. 417–422). Vienna, Austria. [148]

Rodriguez-Serrano, F. J., Carabias-Orti, J. J., Vera-Candeas, P., & Martinez-Munoz, D. (2016). Tempo driven audio-to-score alignment using spectral decomposition and online dynamic time warping. *ACM Transactions on Intelligent Systems and Technology (TIST)*, *8*(2), 22:1–22:20. [14, 15]

Sakoe, H., & Chiba, S. (1978). Dynamic programming algorithm optimization for spoken word recognition. *IEEE Transactions on Acoustics, Speech and Signal Processing*, *26*(1), 43–49. [187]

Salamon, J., & Gómez, E. (2012). Melody extraction from polyphonic music signals using pitch contour characteristics. *IEEE Transactions on Audio, Speech, and Language Processing*, *20*(6), 1759-1770. [85, 86, 87, 229, 240, 250, 277]

Salamon, J., Gulati, S., & Serra, X. (2012). A multipitch approach to tonic identification in Indian classical music. In *Proceedings of 13th International Society for Music Information Retrieval Conference (ISMIR 2012)* (p. 499-504). Porto, Portugal. [248]

Şentürk, S. (2011). *Computational modeling of improvisation in Turkish folk music using variable-length Markov models* (Unpublished master's thesis). Georgia Institute of Technology. [2, 31, 91, 147]

Şentürk, S., Ferraro, A., Porter, A., & Serra, X. (2015). A tool for the analysis and discovery of Ottoman-Turkish makam music. In *Extended Abstracts for the Late Breaking Demo Session of the 16th International Society for Music Information Retrieval Conference (ISMIR 2015)*. Málaga, Spain. [14, 26]

Şentürk, S., Gulati, S., & Serra, X. (2013). Score informed tonic identification for makam music of Turkey. In *Proceedings of 14th International Society for Music Information Retrieval Conference (ISMIR 2013)* (pp. 175–180). Curitiba, Brazil: Pontifícia Universidade Católica do Paraná. [44, 49, 93, 96, 107, 108, 138, 140, 141, 151, 153, 158, 184, 185, 187, 189, 190, 260, 282]

Şentürk, S., Gulati, S., & Serra, X. (2014). Towards alignment of score and audio recordings of Ottoman-Turkish makam music. In *Proceedings of 4th International Workshop on Folk Music Analysis (FMA 2014)* (pp. 57–60). Istanbul, Turkey. [16, 46, 49, 239, 241, 260]

Şentürk, S., Holzapfel, A., & Serra, X. (2012). An approach for linking score and audio recordings in makam music of Turkey. In *2nd CompMusic Workshop* (pp. 95–106). Istanbul, Turkey. [56, 84, 160, 215, 216, 253, 282]

Şentürk, S., Holzapfel, A., & Serra, X. (2014). Linking scores and audio recordings in makam music of Turkey. *Journal of New Music Research*, *43*(1), 34–52. [xlii, xliv, 14, 16, 45, 49, 72, 75, 83, 85, 91, 95, 138, 139, 141, 146, 160, 180, 181, 182, 189, 190, 198, 199, 215, 216, 239, 240, 242, 244, 250, 258, 260, 282]

Şentürk, S., Koduri, G. K., & Serra, X. (2016). A score-informed computational description of svaras using a statistical model. In *Proceedings of 13th Sound and Music Computing Conference (SMC 2016)* (p. 427-433). Hamburg, Germany. [95, 180, 195, 233, 234, 260]

Şentürk, S., & Serra, X. (2016a). Composition identification in Ottoman-Turkish makam music using transposition-invariant partial audio-score alignment. In *Proceedings of 13th Sound and Music Computing Conference (SMC 2016)* (p. 434-441). Hamburg, Germany. [44, 45, 149, 260]

Şentürk, S., & Serra, X. (2016b). A method for structural analysis of Ottoman-Turkish makam music scores. In *Proceedings of*

6th International Workshop on Folk Music Analysis (FMA 2016) (pp. 39–46). Dublin, Ireland. [43, 52, 60, 254, 258]

Serra, J. (1983). *Image analysis and mathematical morphology*. Orlando, FL, USA: Academic Press, Inc. [132, 219]

Serrà, J., Koduri, G. K., Miron, M., & Serra, X. (2011). Assessing the tuning of sung Indian classical music. In *Proceedings of 12th International Society for Music Information Retrieval Conference (ISMIR 2011)* (pp. 263–268). Miami, FL, USA. [240]

Serrà, J., Serra, X., & Andrzejak, R. G. (2009). Cross recurrence quantification for cover song identification. *New Journal of Physics*, *11*(9). [2, 17, 72, 82, 135, 148, 231]

Serra, X. (2011). A multicultural approach in music information research. In *Proceedings of 12th International Society for Music Information Retrieval Conference (ISMIR 2011)* (p. 151-156). Miami, FL, USA. [2]

Serra, X. (2014). Creating research corpora for the computational study of music: The case of the CompMusic project. In *Proceedings of AES 53rd International Conference on Semantic Audio.* London, United Kingdom. [13, 19, 22]

Shetty, S., & Achary, K. (2009). Raga mining of Indian music by extracting arohana-avarohana pattern. *International Journal of Recent Trends in Engineering*, *1*, 362–366. [96]

Signell, K. L. (1986). *Makam: Modal practice in Turkish art music*. Da Capo Press. [7, 8, 10]

Şimşek, B. Ö., Bozkurt, B., & Akan, A. (2016). Fundamental frequency estimation for heterophonical Turkish music by using VMD. In *Proceedings of 24th Signal Processing and Communication Application Conference (SIU 2016)* (pp. 1625–1628). Zonguldak, Turkey. [83, 88, 283]

Six, J., Cornelis, O., & Leman, M. (2014). TarsosDSP, a Real-Time Audio Processing Framework in Java. In *Proceedings of AES 53rd International Conference on Semantic Audio.* London, United Kingdom. [79, 253]

Smith III, J. O., & Serra, X. (1987). *Parshl: an analysis/synthesis program for non-harmonic sounds based on a sinusoidal representation*. CCRMA, Department of Music, Stanford University. [94, 259]

Sordo, M., Chaachoo, A., & Serra, X. (2014). Creating corpora for computational research in Arab-Andalusian music. In *Proceedings of the 1st International Workshop on Digital Libraries for Musicology (DLfM 2014)* (pp. 1–3). London, United Kingdom. Retrieved from `http://doi.acm.org/10.1145/2660168.2660182` [13]

Sordo, M., Koduri, G. K., Şentürk, S., Gulati, S., & Serra, X. (2012). A musically aware system for browsing and interacting with audio music collections. In *Proceedings of 2nd CompMusic Workshop* (pp. 20–24). Istanbul, Turkey. [186]

Srinivasamurthy, A., Holzapfel, A., Cemgil, A. T., & Serra, X. (2016, 20/03/2016). A generalized Bayesian model for tracking long metrical cycles in acoustic music signals. In *41st ieee international conference on acoustics, speech and signal processing (icassp 2016)* (p. 76-80). Shanghai, China: IEEE. [14]

Srinivasamurthy, A., Koduri, G. K., Gulati, S., Ishwar, V., & Serra, X. (2014). Corpora for music information research in indian art music. In *Proceedings of Joint 42nd International Computer Music Conference and 11th Sound and Music Computing Conference (ICMC|SMC|2014)* (p. 1029-1036). Athens, Greece. [13]

Subramanian, M. (2007). Carnatic ragam thodi – pitch analysis of notes and gamakams. *Journal of the Sangeet Natak Akademi*, *XLI*(1), 3–28. [236, 237]

Suma, S. M., & Koolagudi, S. G. (2015). Information systems design and intelligent applications: Proceedings of second international conference india 2015, volume 1. In (pp. 865–875). New Delhi: Springer India. [95, 96]

Swartz, A. (2002). Musicbrainz: A semantic web service. *Intelligent Systems, IEEE*, *17*(1), 76–77. [22]

Tanrıkorur, C. (2011). *Osmanlı dönemi türk musikisi*. Dergah Yayınları. [7, 10, 11]

Tekin, M. E., Anagnostopoulou, C., & Tomita, Y. (2005). Towards an intelligent score following system: Handling of mistakes and jumps encountered during piano practicing. In *Computer music modeling and retrieval: Second international symposium, cmmr 2004, esbjerg, denmark, may 26-29, 2004. revised papers* (pp. 211–219). Berlin, Heidelberg: Springer Berlin Heidelberg. [16]

Temperley, D. (2004). *The cognition of basic musical structures*. MIT press. [59]

Temperley, D., & Marvin, E. W. (2008). Pitch-class distribution and the identification of key. *Music Perception: An Interdisciplinary Journal*, *25*(3), 193–212. [91]

Tenney, J., & Polansky, L. (1980). Temporal gestalt perception in music. *Journal of Music Theory*, *24*(2), 205–241. [59]

Thickstun, J., Harchaoui, Z., & Kakade, S. (2016). Learning features of music from scratch. *arXiv preprint arXiv:1611.09827*. [13]

Thomas, V., Fremerey, C., Müller, M., & Clausen, M. (2012). Linking sheet music and audio - Challenges and new approaches. In *Multimodal Music Processing* (Vol. 3, pp. 1–22). Dagstuhl, Germany: Schloss Dagstuhl–Leibniz-Zentrum für Informatik. [2, 14, 82, 148, 186]

Tomita, E., Tanaka, A., & Takahashi, H. (2006). The worst-case time complexity for generating all maximal cliques and computational experiments. *Theoretical Computer Science*, *363*(1), 28 - 42. [64]

Townsend, M., & Sandler, M. B. (1993). Pattern recognition for formant trajectories using the Hough transform. In *14. colloque sur le traitement du signal et des images* (pp. 1355–1358). Juan-les-Pins, France. [132]

Tura, Y. (1988). *Türk musıkisinin meseleleri*. Pan Yayıncılık, Istanbul (in Turkish). [9, 10]

Tzanetakis, G. (2014). Computational ethnomusicology: a music information retrieval perspective. In *Proceedings of Joint 42nd International Computer Music Conference and 11th Sound and Music Computing Conference (ICMC|SMC|2014).* Athens, Greece. [12]

Tzanetakis, G., & Cook, P. (2002). Musical genre classification of audio signals. *IEEE Transactions on speech and audio processing*, *10*(5), 293–302. [13]

Tzanetakis, G., Ermolinskyi, A., & Cook, P. (2003). Pitch histograms in audio and symbolic music information retrieval. *Journal of New Music Research*, *32*(2), 143-152. [91]

Tzanetakis, G., Kapur, A., Schloss, W. A., & Wright, M. (2007). Computational ethnomusicology. *Journal of interdisciplinary music studies*, *1*(2), 1–24. [2, 12]

Tzanetakis, G., & Lemstrom, K. (2007). Marsyas-0.2: a case study in implementing music information retrieval systems. *Intelligent Music Information Systems. IGI Global*, *14*. [79]

Uyar, B., Atlı, H. S., Şentürk, S., Bozkurt, B., & Serra, X. (2014). A corpus for computational research of Turkish makam music. In *Proceedings of the 1st International Digital Libraries for Musicology Workshop (DLfM 2014)* (pp. 57–63). London, United Kingdom. [13, 19, 20, 31, 108, 154]

van der Walt, S., Colbert, S. C., & Varoquaux, G. (2011). The NumPy array: A structure for efficient numerical computation. *Computing in Science & Engineering*, *13*(2), 22-30. [255]

van Kranenburg, P., Janssen, B., & Volk, A. (2016). *The meertens tune collections: The annotated corpus (mtc-ann) versions 1.1 and 2.0.1.* Meertens Online Reports. [13]

Wang, A. L.-C. (2003). An industrial strength audio search algorithm. In *Proceedings of 4th International Society for Music Information Retrieval Conference (ISMIR 2003)* (pp. 713–718). Baltimore, MD, USA. [2]

Yarman, O. (2008). *79-tone tuning & theory for Turkish maqam music* (Unpublished doctoral dissertation). İstanbul Teknik Üniversitesi Sosyal Bilimler Enstitüsü. [8]

Çevikoğlu, T. (2007). Klasik türk müziğinin bugünkü sorunları. In *Proceedings of International Congress of Asian and North African Studies (Icanas 38').* Ankara, Turkey. [27]

Index