UAB

Universitat Autònoma
de Barcelona

Focused Structural Document Image Retrieval in Digital Mailroom Applications

A dissertation submitted by **Hongxing Gao** at Universitat Autònoma de Barcelona to fulfil the degree of **Doctor en Informàtica** de **Dep. de Ciéncies de la Computació**.

Bellaterra, November 2014

Directors	**Dr. Josep Lladós** Centre de Visió per Computador Dept. de Ciències de la Computació, Universitat Autònoma de Barcelona **Dr. Dimosthenis Karatzas** Centre de Visió per Computador Dept. de Ciències de la Computació, Universitat Autònoma de Barcelona **Dr. Marçal Rusiñol** Centre de Visió per Computador Dept. de Ciències de la Computació, Universitat Autònoma de Barcelona
Thesis Committee	**Dr. David Doermann** Language and Media Processing Laboratory University of Maryland at College Park **Dr. C.V. Jawahar** Internaional Inst. of Information Technology **Dr. Andrew Bagdanov** Centre de Visió per Computador Universitat Autònoma de Barcelona **Dr.Alcia Fornes** Centre de Visió per Computador Universitat Autònoma de Barcelona **Dr. Cristina Caero Morales** ICAR Vision Sytems
European Mention Evaluator	**Joseph Chazalon** Laboratoire Informatique, Image et Interaction (L3i) Université de La Rochelle **Antoine Tabbone** Laboratoire Lorrain de Recherche en Informatique et ses Applications (LORIA) Université de Lorraine

This document was typeset by the author using $\LaTeX\,2_\varepsilon$.

The research described in this book was carried out at the Computer Vision Center, Universitat Autònoma de Barcelona.

To my family,

and all those who made a better me.

Acknowledgments

I would like to especially express my gratitude my supervisors Dr. Josep Lladós, Dr. Dimosthenis Karatzas and Dr. Marçal Rusiñol who gave me numerous of suggestions and shared lots of great ideas that drive our research forward. Trough the whole way of my research, it is them who patiently helped me on every problem and enthusiastically offering their knowledge and wisdom. Without their countless guidance, advises and help along the way, I would never make this research possible. Thanks to their consideration for improving my abilities, I was given precious opportunities to collaborate with peers and professors in other institutes. Among those advisable chances, I would especially appreciate the cooperation with Dr. Rajiv Jain from Language And Media Processing (LAMP) Laboratory of University of Maryland. Besides, I would also like to thank all other members in the Document Analysis Group in Computer Vision Center (CVC). They generously shared great number of ideas for document analysis challenges and inspired me to solve the problems in my research.

Other than the above peoples who made significant contributions for my research, I would also like thank Chinese Scholarship Council and Universitat Autónoma de Barcelona for awarding me the scholar (No.2011674029) for covering the living expenses and tuition fees. Besides, my particular appreciations go for the Spanish project that offers the funding for attending many impressive international conferences, actively participating the summer schools and also performing a precious fruitful research stay in LAMP laboratory. Those impressive activities helped me to exchange the ideas with the people working in various areas and thus significantly widen my vision in document analysis field and gave me lots of inspirations on my research.

I would also express my gratitude for Dr. David Doermann, Dr. C.V.Jawahar and Dr. Andrew Bagdanov for making their grateful effort for reviewing my thesis and spending their valuable time to travel to CVC as my thesis committee. I am also very thankful to my European Mention Evaluators Dr. Joseph Chazalon and Dr. Antoine Tabbone Besides, I gratefully acknowledge the one who lead my way and helped me in the middle: my supervisor during my master study, Prof. Yigong Peng from East China University of Science and Technology, who guided me heading to a restrict and honest researcher and gave me lots of support preparing the application for PhD student position. I would also thank those who assisted me to prepare countless documents for applying the scholarship and legally staying here, including but not limited to Zhonghua Deng from the International College of Engineering of East China University of Science and Technology, Montse Culleré, Claire Perez-Mangado

and Gisele Kohatsu from CVC.

Furthermore, I will always owe my thanks to my dearest family and best friends. My father Qingwen Gao, my mother Xiuying Zhang, my brothers Liangliang Gao, Changqing Gao and my sister Yanqiu Gao, who gave me their endless love and warmest support for my homesick. I am also very grateful for my girlfriend Nuria Cirera Turigues and her father Miquel Cirera Roca, her mother Maria Assumpció Turigues Vernet, her aunt Maria ngles Turigues Vernet and her grandfather Miquel Turigues Vinyes who shared every moment of my life in Barcelona Catalunya during the time of both happiness and sadness. Besides, I would also like thank my friends who shared numerous pleasant moments and delicious dinners during my PhD study including Jiaolong Xu, Zhanwu Xiong, Lluís Gómez, Wenjuan Gong, Xu Hu, Marc Serra, Patricia Máquez, Peng Wang, Anjan Dutta, my roommates Huang Zhang, Hua Feng, Yangchun Xin, Peng Zhang, my friends in China Tiantian He, Juan Xie, Xinxing Yu, Ru Wang, Qinqin Fan, Wenwen Hou and those who gave a helping hand but not listed here.

Abstract

In this work, we develop a generic framework that is able to handle document retrieval problem in various scenarios such as searching the full page matches or retrieving the counterparts for specific area of document, focusing on their structural similarity or letting their visual feature to play the dominant role. Based on the spatial indexing technique, we propose to search matches for the local key-region pairs carrying both structural and visual information from the collection while a scheme to adjust the structural and visual similarity is presented.

Based on the fact that the structure of documents is tightly linked with the distance among their elements, we firstly introduce an efficient detector named as Distance Transform based MSER (DTMSER). We illustrate that it is able to extract the structure of a document image as a dendrogram (hierarchical tree) of multi-scale key-region that roughly correspond to letters, words, paragraphs. We demonstrate that, without benefiting from the computed structure information, the key-regions extracted by DTMSER algorithm achieves slightly better result comparing with state-of-the-art methods (SIFT, MSER) while much less amount key-regions are employed.

We subsequently propose a pair-wise BoW framework to efficiently embed the explicit structure lies within the dendrogram extracted by DTMSER algorithm. We represent each document as a list of key-region pairs that correspond to the edges in the dendrogram where *inclusion* relationship is encoded. By employing those structural key-region pairs as the pooling elements for generating the histogram of features, we demonstrate that the proposed method is able to encode the explicit *inclusion* relations into BoW representation. Besides, we apply the inverted file indexing techniques to solve the quadratic computation complexity problem inherited from the pair-wise representation. The experimental results illustrate that the pair-wise BoW, powered by the embedded structural information, achieves remarkable improvement over the conventional BoW and spatial pyramidal BoW methods.

To handle various retrieval scenarios in one framework, we propose to directly query a series of key-region pairs, carrying both structure and visual information, from the collection. We introduce the spatial indexing techniques into document retrieval community to speed up the structural relationship computation for key-region pairs. We demonstrate that the proposed framework allows to adjust the role that the structure and visual features would play when measuring the similarity through tuning the discriminative power of the two types of visual features (geometrical and content). We firstly test the proposed framework in a full page retrieval scenario where the structural similar matches are expected. In this case, the pair-wise querying method achieves notable improvement over the BoW and spatial pyramidal BoW frameworks. However, slight performance decrease is observed comparing with the pair-wise BoW method due to the ”noise” introduced by the spatial indexation. Furthermore, we illustrate that the propose method is also able to handle the focused retrieval situations where the query is defined as a specific interesting area of the image such as logos, address blocks or shopping records. We perform our method on two types of focused queries: *structure-focused* queries that search their counterparts by the structural similarity where the content variation is allowed and *exact* queries that look for the matches hold similarity on both structure and their visual content. The experimental results show that, the proposed generic framework obtains nearly perfect precision on both type of focused queries while reasonable lower recall is observed for *structure-focused* queries. It is because RANSAC algorithm fails to find lots of matches from the collection for the *structure-focused* queries when searching multi instance in one image due to the large portion of outliers.

To solve the problem of RANSAC which is observed to be too rigid for *structure-focused* queries, we introduce a line verification method as an alternative strategy to check the spatial consistency among the matched key-region pairs. We illustrate that the line verification is more robust over outliers. Since it is very expensive to compute the lines for all combination of two points, we propose a cheaper version through a two step implementation. We first compute tentative bounding boxes that might be not precise but can be employed to divide the matched key-region pairs into several groups, then line verification is performed within each group and compute the corresponding bounding boxes more precisely. We demonstrate that, comparing with RANSAC, the line verification generally achieve much higher recall with slight loss on precision on specific queries.

Resumen

En la presente investigacin se desarrolla un marco de trabajo genrico para la bsqueda de documentos digitales partiendo de un documento de muestra, tanto para solicitudes de imagenes completas como subpartes de la misma, donde el criterio de similitud hace uso tanto del parecido a nivel estructural como de otras caracteristicas visuales relevantes. Partiendo de la tcnica de indexacin espacial proponemos la utilizacin de correspondencias entre pares de regiones locales de inters, aportando estas informacin tanto estructural como visual, y al mismo tiempo detallamos un metodo para definir la combinacin de ambos tipos de informacin en un unico criterio de similitud.

Partiendo del hecho constatado que la estructura de un documento est intrnsecamente ligada a las distancias que definen su contenido, primeramente presentamos un detector eficiente que bautizamos como DTMSER, basado en el ya existente MSER y modificado con la transformada de la distancia. Mostramos que este detector es capaz de extraer la estructura del documento en forma de dendograma (arbol jerarquico) de regiones de inters a diferentes escalas, las cuales corresponden aproximadamente a caracteres, palagras y prrafos. Los experimentos realizados prueban que el algoritmo DTMSER logra mejores resultados respecto a mtodos consagrados como SIFT y MSER, con la ventaja de usar menos regiones de inters que dichos mtodos de referencia.

Posteriormente proponemos un metodo basado en pares de descriptores BoW que permite representar el dendograma extraido mediante el algoritmo DTMSER. Para este fin cada documento se representa mediante una lista de pares de regiones de inters, donde cada par representa una arista del dendograma y define la relacin de inclusin entre ambas regiones. Dado que el histograma de caracteristicas es generado en base a tales pares de regiones de inters y los resultados obtenidos son satisfactorios, se demuestra que el metodo propuesto refleja fielmente las relaciones de inclusin de regiones. Tal metodo requerira un tiempo de computacin cuadratico debido al uso de pares de regiones, pero mostramos como solventarlo mediante tecnicas de indexacin inversas de ficheros. Los experimentos realizados demuestran que el metodo propuesto supera con creces otras variantes de BoW, tanto convencionales como de espacio-piramidales.

Teniendo en cuenta diferentes situaciones donde se puede requerir la busqueda de documentos digitales, proponemos aglutinar estas situaciones y trabajar directamente partiendo de pares de regiones de inters, donde se incluye informacin espacial y tambien visual. Para ello usamos en este campo tecnicas de indexacin espacial para agilizar los calculos relativos a la similtud entre pares de regiones. El marco de trabajo propuesto permite encontrar un equilibrio satisfactiorio entre caractersticas estructurales y caractersticas visuales. En primer lugar aplicamos este marco de trabajo al caso comn de bsquedas de pginas enteras, donde se espera que los resultados presenten una gran similitud espacial respecto a la imagen inicial. En este experimento nuestro metodo de bsqueda basado en pares de regiones supera la mayoria de mtodos BoW analizados. Aqui el uso de la indexacin inversa de ficheros no juega a nuestro favor debido al ruido que introduce la busqueda de pares de regiones de inters, y como resultado el metodo de referencia BoW dos a dos supera ligeramente el nuestro. Sin embargo nuestra propuesta es claramente superior ya que permite buscar partes de documento tales como logotipos, direcciones postales, listados en albaranes, etc. Aplicamos esta bsqueda focalizada a dos casos distintos: priorizando la *similitud estructural* pero permitiendo diferente contenido, y la ms restictiva de *estructura y apariencia similares* donde adems se requiere un contenido similar. Los resultados obtenidos son excelentes para ambos casos, aunque el primer caso presenta menor sensibilidad debido al efecto de gran cantidad de valores atpidos en el algoritmo RANSAC.

Para las busquedas con *similitud estructural* proponemos solucionar el problema del algoritmo RANSAC mediante un metodo de verificacion de lineas, frente al genrico punto a punto de RANSAC en la verificacin de pares de regiones de inters. Mostramos la robustez de nuestro metodo de comparacin de lneas en presencia de gran cantidad de valores atpicos. Para aligerar la carga computacional de nuestro metodo definimos una simplificacin practica en tres pasos. El primer paso es obtener candidatos a regiones de inters para luego posteriormente realizar la verificacin de lineas en estas y finalmente precisar las regiones de inters. Los experimentos demuestran que nuestra propuesta de verificacin de lineas es ms exhaustivo respecto a RANSAC a expensas de un ligero decremento en precisin, lo cual es preferible en determinados casos de busqueda.

Resum

Aquesta tesi doctoral presenta un marc de treball genric per a la cerca de documents digitals partint d'un document de mostra de referencia, on el criteri de similitud pot ser tant a nivell de pgina com a nivell de subparts d'inters. Aquest criteri de similitud conjuga les relacions estructurals entre regions del document aix com caracteristiques purament visuals. Combinem la tecnica d'indexaci espacial amb correspondncies entre parells de regions locals d'inters, on aquestes contenen informaci tant estructural com visual, i detallem la combinaci adient usada d'aquests dos tipus d'informaci per ser usada com a nic criteri de similitud a l'hora de fer la cerca.

Donat que l'estructura d'un document est lligada a les distncies entre els seus continguts, d'entrada presentem un detector eficient que anomenem DTMSER, basat en el ja existent MSER pero modificant-lo amb la transformada de la distncia. El detector proposat s caps d'extreure l'estructura del document en forma de dendograma (arbre jerrquic) de regions d'interes a diferents escales, les quals guarden una gran similitud amb els caracters, paraules i pargrafs. Els experiments realitzats proven que l'algorisme DTMSER supera els metodes de referncia SIFT i MSER, amb l'avantatge de requerir menys regions d'interes.

A continuaci proposem un mtode basat en parells de descriptors BoW que permet representar el dendograma descrit anteriorment i resultat de l'algorisme DTMSER. El nostre mtode consisteix en representar cada document en forma de llista de parelles de regions d'inters, on cada parella representa una aresta del dendograma i defineix una relaci d'inclusi entre ambdues regions. Com que l'histograma de caracterstiques s generat a partir de les prelles de regions d'inters i els resultats sn satisfactoris, podem concloure que el mtode proposat reflecteix la inclusi de regions. Aquesta proposta requeriria un temps de clcul quadrtic degur a l's de parells de regions, pero tamb incloem l'opci d'usar tcniques d'indexaci inversa de fitxers per alleugerir aquesta crrega. Els experiments realitzats demostren que el nostre mtode supera mpliament altres variants exteses de BoW com poden ver les convencionals o les espacio-piramidals.

Per tal d'englobar diferents situacions on es pot requerir una la cerca de documents digitals, proposem usar directament parelles de regions d'inters, les quals inclouen informaci tant espacial com visual. Amb aquest objectiu introduim en aquest camp tcniques d'indexaci espacial per millorar el temps de clcul de les similituds de parelles de regions. El marc de treball que porposem permet definir satisfactriament la combinaci equilibrada entre les caracterstiques estructures i les visuals. Apliquem la nostra proposta al cas de cerques de pgines senceres, on t ms pes la similitud estructural. Els experiments corresponents mostren que la nostra proposta supera la majoria de mtodes BoW de referncia. En aquestes condicions l's d'indexaci inversa de fitxers no millora la cerca ja que afegeix soroll en la cerca de parelles de regions d'inters, i per aquest motiu el mtode de referencia BoW dos a dos supera lleugerament la nostra proposta. No obstant aix, la nostra proposta presenta un clar avantantge: podem fer cerques de subparts de documents, com serien logotips, blocs de direccions postals, llistes d'albarans, etc. Apliquem el nostre metode en la cerca de subparts en dos casos: prioritzant la *similitud estructural* per permetent diferncies de contingut, i restringint la cerca unes *estructura y aparena similars* on ara el contingut ha de ser semblant. Els resultats obtinguts en els experiments sn excellents en tots dos casos, tot i que el primer cas l'algorisme RANSAC usat presenta menys sensibilitat en presncia de grans quantitats de valors atpics.

Per millorar les cerques amb *similitud estructural* proposem la millora de l'algorisme RANSAC mitjanant la verificaci de lnies, substituint el genric punt a punt de RANSAC a l'hora de verificar parells de regions d'inters. Mostrem la robustesa de la nostra proposta de comparaci de lnies en presncia de valors atpics. Per reduir la crreca computacional de la nostra proposta definim una simplificaci practica en tres passos. Primer obtenim candidats a regions d'inters per posteriorment realitzar la verificaci de lnies en aquestes regions, i finalment precisem les regions d'inters. Els experiments demostren que la nostra roposta de verificaci de lnies es ms exhaustiva que RANSAC a canvi de permetre una disminuci de precisi, la qual cosa es preferible en determinats casos de cerca.

Contents

Acknowledgments i

Abstract iii

Resumen v

Resum vii

1 Introduction 1
1.1 Problem Definition 2
1.2 Motivation 3
1.3 Challenges 4
1.4 Our Contributions 5

2 State of Art 7
2.1 Layout Analysis based Retrieval 8
2.1.1 Document Segmentation 8
2.1.2 Block Matching 9
2.2 Local Content Feature based Retrieval 10
2.2.1 Local Representation 10
2.2.2 Global Representation 11
2.2.3 Encoding Structure for Local Features 12

3 Real-time Pyramidal Document Structural retrieval 15
3.1 Global Image Description 15
3.2 Retrieval by Similarity 16
3.3 Relevance Feedback 18
3.3.1 Rocchio's Algorithm 19
3.3.2 Relevance Score 20
3.4 Experimental Results 20
3.4.1 Dataset and Evaluation Measures 20
3.4.2 Evaluation of different distance metrics 21
3.4.3 Results on Relevance Feedback 23
3.5 Summary 25

4 Distance Transform based MSER 27
4.1 Key-Region Detectors in Document Analysis . . . 30
4.2 Distance Transform based MSER . . . 30
4.2.1 Distance Transform . . . 31
4.2.2 MSER detection . . . 32
4.3 Experimental Results . . . 33
4.3.1 Qualitative results . . . 34
4.3.2 Comparative results on a subset . . . 34
4.3.3 Results on the whole dataset . . . 35
4.4 Conclusions . . . 37

5 Pair-wise BoW 39
5.1 Feature Extraction . . . 42
5.1.1 Distance based MSER (DTMSER) . . . 43
5.1.2 Descriptors . . . 44
5.1.3 Codebook . . . 45
5.1.4 Pair-wise Pooling . . . 46
5.2 Inverted File Indexing . . . 46
5.3 Experiments . . . 47
5.3.1 Parameters Validation . . . 47
5.3.2 Proposed VS BoW . . . 49
5.4 Conclusion . . . 50

6 Spatial Database 53
6.1 Pair-wise spatial retrieval . . . 55
6.1.1 Document representation . . . 55
6.1.2 Spatial Indexing . . . 57
6.1.3 Structural Retrieval . . . 58
6.2 Experiments . . . 59
6.2.1 Full Page Document Image Retrieval . . . 59
6.2.2 Focused Retrieval . . . 64
6.3 Conclusion . . . 67

7 Spatial Verification 69
7.1 Spatial Verification . . . 69
7.1.1 RANSAC algorithm . . . 70
7.1.2 Geometrical Verification . . . 70
7.1.3 Proposed Line Verification . . . 72
7.2 Experiment Results . . . 73
7.3 Conclusion and Future Work . . . 76

8 Conclusion and Future Work 79

A Datasets 83
A.1 Full page retrieval . . . 84
A.2 Focused Retrieval . . . 85

B Euclidean and Cosine Distance **87**

C Publications **89**

Bibliography **91**

List of Tables

3.1 Average MAP of all the queries. 21
3.2 Time Consumption of Feature Extraction and Distance Calculation. . 23
3.3 Average MAP of all the queries. 24

4.1 MAP and time consumption for sub dataset 35
4.2 MAP time consumption for whole dataset 37

5.1 MAP performance of descriptors and frameworks(n_geom = 25 and $n\_des = 200$) 49

6.1 Data Structure Stored in Spatial Database. 58
6.2 MAP performance of descriptors and frameworks(n_geom = 20 and $n\_des = 250$) 62
6.3 Detailed performance on *Structure-focused* queries 65
6.4 Detailed performance on *Exact* queries 66
6.5 Performance on part-based queries (codebook size fixed) 67

7.1 Performance on focused queries whereas the structural similarity is especially focused (Q1 Q10). 75
7.2 Performance on part-based queries (codebook size fixed) 75

A.1 Number of true positives of the queries for focused retrieval. 86

List of Figures

1.1 Document Retrieval in 3 different scenario: 1) Full page retrieval that searching for both content and structural similar images 2)Exact Matches aiming to retrieve all the image parts that preserve the content and structural similarity 3)Structure-focused matches is expected to return the image parts that the structural similarity is preserved within which the content could be varied 2

3.1 Demonstration of Pyramidal Decomposition. The extracted feature vector is showed at the bottom. 16
3.2 Demonstration of Relevance Feedback. The documents with green frames are the true relevant images while the images with red frames are falsely recognized as relevant. As shown here, with one iteration of relevance feedback, the MAP performance increases from 0.7257 to 0.9308 19
3.3 Precision-Recall curve of 500 queries (both front-Pages and non-front-Pages, L5 feature vector). 22
3.4 MAP improvement by Relevance Feedback. 24

4.1 Typical applications of document retrieval include historical and administrative document analysis uses. (a) images taken from the IMPACT historical newspapers dataset, where a typical application is the retrieval of front pages - used with permission, (b) images from a typical digital mailroom page flow, a typical application being the retrieval of invoices from the same provider. 29
4.2 Demonstration of the distance transform. a) Original image, b) its distance transform 31
4.3 Example of thresholding the distance transform at different intensity levels. At the lower level individual characters can be identified, at the middle level characters have been merged into words, at the upper level words have been merged into paragraphs. 32
4.4 A typical dendrogram produced with the proposed method. The leaf nodes correspond to the connected components of the image, while the mergers depend solely on the distance between regions, giving rise to semantically relevant groups. 33

4.5 Qualitative comparision of a) SIFT, b) MSER and c) DTMSER key-region detectors. 34

4.6 Precison-Recall curve for the sub-dataset when using the truncated inverse distance scoring method. 36

5.1 The pipeline of BoW representation for document images. 40

5.2 Encoding spatial structure into BoW representation 41

5.3 The pipeline of document retrieval based on the proposed pair-wise representation. 42

5.4 Progress of Distance Transform MSER algorithm.(a) Distance transform; (b) Thresholding process for MSER analysis; (c) the resulted key-regions represented as ellipsoids and a dendrogram linking the key-regions together. 43

5.5 Pair-wise BoW pipeline with details. 45

5.6 The Inverted File Indexing structure of the key-region pairs. 47

5.7 Clustering parameter Validation of a) Run Length, b) SIFT and c) HOG descriptor based on BoW. 48

5.8 Clustering parameter Validation of a) Run Length, b) SIFT and c) HOG descriptor based on Pyramidal BoW 48

5.9 Clustering parameter Validation of a) Run Length, b) SIFT and c) HOG descriptor based on the proposed method. 49

5.10 Precision-Recall curve of SIFT descriptor based on BoW, Pyramidal BoW and the proposed method. 50

6.1 Document Retrieval in three different scenarios: (a) Exact matches aiming to retrieve all the image parts that preserve the content and structural similarity; (b) Full page retrieval searching for both content and structurally similar images; (c)Structure-focused matches expected to return the image parts where the structural similarity is preserved but the content could be differ. 54

6.2 Pipeline of the proposed structural matching. 56

6.3 R-tree indexing for spatial database. 57

6.4 Image coordinate definition. 58

6.5 Illustration of the Top10 results of the proposed framework for full-page retrieval. 60

6.6 Illustration of the Top10 results of the proposed framework for structure-focused retrieval. 61

6.7 Illustration of the Top10 results of the proposed framework for exact retrieval. 61

6.8 Clustering parameter validation of the proposed method: a) Run Length, b) SIFT and c) HOG descriptor based on Spatial Database framework. 62

6.9 Precision Recall curve of SIFT descriptor based on BoW, Pyramidal BoW and the proposed method. 63

6.10 MAP performance for *Structure-focused* queries in terms of clustering configuration and codebook size respectively 65

6.11 MAP performance for *Exact* queries in terms of clustering configuration and codebook size respectively. 66

7.1 Geometrical verification by checking the consistency between triangles. 71
7.2 The example of matched points (key-region pairs in our case). The blue bounding boxes correspond the key-region pairs while the big red and black ones indicate the focused query part and its expected matches respectively. Only 20% of the overall matches are shown here for a nice visual. 72
7.3 Geometrical verification by check the relations between lines and estimate the transformation for the matched points. The dotted lines show the mapping relation of lines. The blue dotted lines correspond to correct transformation estimation and the red dotted line indicate the wrong ones. 72
7.4 Two step line verification. 74
7.5 Two step line verification. 76
7.6 Two step line verification. 76

A.1 The statistical distribution of the size of all the invoice classes. 83
A.2 Samples of invoice images from four providers. 84
A.3 Samples of the structured focused queries. 85
A.4 Samples of the queries that seek for exact matches from the collection. 86

B.1 MAP improvement by Relevance Feedback. 88

Chapter 1

Introduction

During the past decades, considerable amount of document databases have been created since the digitalization of the document images is becoming to be more convenient and cheaper. For example, instead of expensive professional scanners, one could upload images nowadays through digital camera or even smart-phone which is ubiquitous in our life. However, how to access and analyze the massive information at low cost time is still a challenging problem. Depending on the types of content that the users concern, many techniques have been developed for many applications such as spotting specific words, validating the signature for security purpose and classifying the stored data according to the writers for hand-written document images etc. Among the various scenarios, document image retrieval receives significant attention among the document image analysis domain and successfully applied to our really life. For instance, one would probably want to efficiently collect all the administrative forms from the same provider by matching the logos or business/private letters from the same sender according to the titles or address blocks. As the smartphone is becoming affordable for most people, similar or exact document could be retrieved from a large number of images by taking a single photo the query in nearly real-time. Similar situations may also happen to newspapers, mails, journals, magazines, books etc.

Despite that enormous research on document image retrieval has been done for various situations, there are still many challenges that have not unsolved yet. However, exploiting document structure in a stable manner is still missing while the layout-based methods are retarded due to the segmentation errors and matching complexity. As increasing number of images are keep adding into the database, the efficiency of image storing, feature extraction and matching is also still an open problem. Another concern for document image retrieval is that if the system is capable to return part of database images that match the query best or the system only can search document at full-page level. Besides, depending on specific applications, users may carry different conception/criteria on similarity among images such as structure similar or content similar etc. Even though lots of methods have been demonstrated to be effective for some specific scenarios, a generic framework aiming at solve the whole spectrum of document retrieval problem have not been exploited.

1.1 Problem Definition

We can frame the problem of document image retrieval as returning a list of boundingboxes $\{bbox_1, bbox_2, \ldots, bbox_i, \ldots, bbox_n\}$ which has been sorted according to corresponding score that measuring the similarity of query image and the matched target image as follows:

$$Score_i = f(Sim(Query, bbox_i)) \tag{1.1}$$

while $bbox_i$ could further formulated as $bbox_i = \{im\_index, x, y, width, height\}$ where im_index is employed to indicate the image that the boundingbox located from the database and $\{x, y, width, height\}$ represent the coordinate of top-left points and the width and height of the boundingbox specifying part of the image. Even though the queries and matches are rectangle zones in most cases, we should note that for some specific situations, the boundingbox we mentioned here could be substituted by polygons or circles.

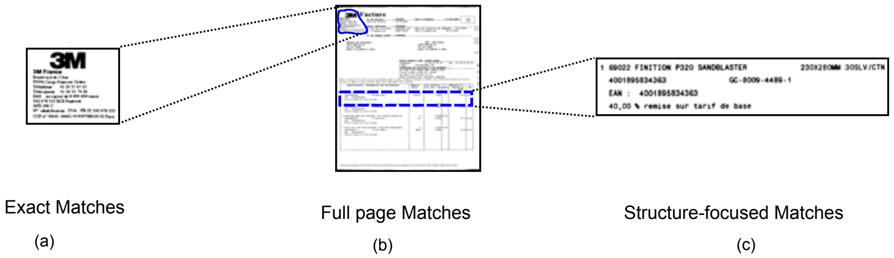

3M
1 69022 FINITION P320 SANDBLASTER 230X280MM 30SLV/CTN
4001895834363 GC-8009-4489-1
EAN : 4001895834363
40,00 % remise sur tarif de base
Exact Matches
(a)
Full page Matches
(b)
Structure-focused Matches
(c)

Figure 1.1: Document Retrieval in 3 different scenario: 1) Full page retrieval that searching for both content and structural similar images 2)Exact Matches aiming to retrieve all the image parts that preserve the content and structural similarity 3)Structure-focused matches is expected to return the image parts that the structural similarity is preserved within which the content could be varied

As shown in Figure 6.1, document image retrieval could be categorized into two types: 1) full-page matching that searching whole document as the final target and thus calculate the similarity score between query image and the complete target image; 2)part-based document retrieval that try to find the part of target image that hold high similarity score with query image. For instance, searching the whole page captured from checks, books, journals or magazines could be considered as the full page retrieval scenario. On the other hand, searching address block, shopping records, company logos could be categorized as part-based retrieval problem. For the full-page retrieval problem, the bounding box applied in (1.1) should be forced to the outline of the document while, for part-based retrieval, the bounding box could be located anywhere inside the target document. Besides, in some applications, the user may look for multiple instances (bounding boxes) from one single image.

Besides, from the viewpoint of similarity, document image retrieval could be divided into many categories where content-based retrieval and structure-based retrieval are the most common ones. The content-based retrieval scenario aims at searching the matches that hold high similarity on both content (and also the structure sometimes) between images while the structure-based querying does not concern much on the content but pays more attention on structure similarity. For example, when searching the documents that contain the given logo that looking for the exact counterpart or with few changes in some cases, the similarity on both structural characteristic and content feature would be considered. However, when the user want to get all the shopping records from one provider, it might be better to compute the similarity score based on the structure only since the conveyed contents may vary on product name, quantity, price and so on.

In summary, our research aims at generic framework that could solve the whole spectrum of document retrieval purpose from full-page retrieval to image part searching, from content-based matching to structure-based querying. Even though significant amount of research have been done for solving the problem from different aspect, there are few algorithms that are capable to generally deal with the various demands in one framework.

1.2 Motivation

As discussed before, document image retrieval problem yields to many scenarios such as full-page and part-based matching where the similarity might be measured on content, structure or both of them. Considerable works have been done that focused on their own specific applications and achieve good performance for the given situations. However, few methods could generally handle the whole series of retrieval scenarios. However, in reality, the users might seek for multi-objective solution in one single framework to generally handle many different problems. For example, in most cases, the administrative employee may use document analysis system to categorized the invoices by matching the provider's logo which could be considered as image part retrieval problem based on the content and structure similarity. However, they may appreciate that if the same system could also fetch the original full document image from the database in the case that part of the printed version is destroyed due to coffee stain or unintentional doodling. Besides, from the viewpoint of commercial cost, space usage and system setup, the system that is capable to generally deal with multi problems might be more economical than buying many systems each of which only focus on limited problems.

When measuring the similarity of query and the counterparts, the most straightforward way is to check the difference of the conveyed contents based on either OCR-ed strings or local key-points (key-regions) features. However, in most situation, the document structure, typically represented as either the spatial relation between segmented blocks or the hierarchical tree of the document elements, also plays significant role in evaluating the similarity between query and target images. Document layout analysis is the most straightforward and popular way for analyzing the document structure. The layout analysis methods usually segment the document into group of blocks and represent the structure as the spatial relation among the segmented blocks. However, apart of the stability issue on segmentation that is still an open challenge, the high computation cost for measuring the similarity between a group of blocks and another also hinders the applications of layout analysis methods. Another popular manner for representing document structure is to encode the location information of key-points (key-regions) into feature vector through dividing the image into pyramidal parts. Even though such spatial structure encoding methods achieves nice performance, the structural information that actually encoded in the feature vector is implicit local patterns rather than explicit low-level structure and does not yield to image rotation. Consequently, it is appealing to exploit one method to extract the document structure in stable and repeatable manner where the content feature and structure information could be combined together to evaluate the similarity between query and target images.

Another aspect to be specially concerned for document image analysis problem is the efficiency. This consists of the time consumption of content feature extraction, the structure representation and distance computation for both content and structure feature, sometimes the spatial consistence checking if applicable. Hence, the efficient feature detection and description as well as the indexing strategy of both feature vector and spatial relations for structure are also very attractive.

1.3 Challenges

As discussed before, document image retrieval problem yields to many different scenarios such as full-page or image part querying, content concentrated or structure-based similarity for various document type like newspapers, letter, books, invoices etc. Generally representing document for various situation in one single framework is extremely difficult because different scenarios may have varied demands and criteria of similarity. However, the fact is that the common characteristics that most documents share are that they are composed by elements such as letters, words, paragraphs (figures and tables if applicable) which are linked together in the same way: letters to words, and words to paragraphs. These common characteristics imply the possibility of a general framework for dealing document images in various scenarios.

One common issue for document image analysis, and also for natural scene images, is the feature extraction problem which is very challenging to meet the discriminative and repeatable requirement. Various feature detection methods such as Scale Invariant Feature Transform (SIFT) and Maximal Stable Extremal Region (MSER) have been proposed for natural scene understanding and achieve remarkable success in document analysis domain as well. However, the current detection methods might not be able to naturally extract the semantical elements of the document. For example, in the situation of binary images, SIFT detects the blobs that basically correspond to corners and edges of letters and MSER algorithm extracts the connected components that correspond to letters only in most case. On the other hand, layout analysis methods representing document features as segmented blocks are still suffering from the segmentation errors and the matching complexity. Consequently, extract the document elements at different level in an efficient manner is still another open challenging problem.

Besides the content contained in the document, the structure is another rich source of image description many works have been studied for document structure extraction for some specific types of documents. Due to the huge variation among documents, representing their structure in an efficient and stable way is still a challenge. Moreover, indexing the document structure which normally represented as graph is also challenging even though many methods like graph embedding, random walk etc. have been studied to computed the distance between graphs.

1.4 Our Contributions

To solve the challenging problems listed in the previous section, we made minor step forward for document image analysis as follows:

Based on the fact that the letters usually are placed closer to each other than words are and in turn the words are closer than paragraphs do, we propose Distance Transform based MSER (DTMSER) detection method which performs MSER analysis process on distance transformed image where the value of each pixel is set to corresponding distance to the nearest object/foreground pixel. DTMSER detection algorithm is capable to extract the key-regions at different semantical levels such as letters, words and paragraphs. On contrast, SIFT basically extracts corner or edge blobs and MSER simply returns connected component. Besides, we demonstrate that the content features extracted from DTMSER key-regions hold better discriminative power than the features extracted from SIFT key-points and MSER key-regions for binary document images scenario. Moreover, the main advantage of DTMSER algorithm is that it could explicitly extract the document structure in an efficient manner as a dendrogram (hierarchical tree) that roughly defines how letters merger to words, words to paragraphs and paragraphs to the whole document.

Besides, base on Bag-of-Words (BoW) algorithm, we propose an efficient framework to embed document structure into BoW histogram through key-region pairs where labels are assigned according to the content features of corresponding key-regions. Comparing with the standard BoW method that proposed for natural scene image understanding and also achieves great success in document image analysis afterwards, the proposed framework efficiently encode the explicit *inclusion* structural relationships among the DTMSER key-regions into the representation vector while the standard BoW method only encode the orderless separated key-regions. Besides, to handle structure-based retrieval applications, which allows the text variation on content, we propose to employ the geometrical feature that is more robust on variation of the conveyed content together with the content features.

We further introduced spatial database into document analysis domain to spatially index the document structure to efficiently query structural relations among key-regions. The DTMSER key-regions are stored in spatial database in terms of corresponding bounding boxes based on which the spatial indexes are built. Such spatial indexation strategy significantly reduce the time consumption for explicitly structure querying such as "return all key-region pairs where one labeled as **A** lies within another labeled as **B**". Taking the advantage of the advanced spatial indexing techniques that developed for Geography Information System (GIS), our system is capable to handle many different structural relationships between key-regions such as one is placed of $right/left$ or *top/bottom* of another. In the case of DTMSER key-regions, the most straightforward and important one for most cases is the *inclusion* relation between the extracted key-regions. We illustrate that the proposed spatial indexing framework could be applied to both full-page and image part document retrieval scenarios. Moreover, apart of the content features of key-regions, the assigned label also implicitly encodes the information of geometrical features that suit for structure-based retrieval scenarios. Consequently, the proposed document retrieval system is a general solution that is capable to handle both content-based and structure-based searching for both full-page and part-based retrieval situations.

In order to solve the over-rigid problem of RANSAC algorithm that applied for checking the spatial consistency of the matched key-region pairs, we further studied the line verification method as an alternative solution of RANSAC. Different with the global spatial consistency that RANSAC algorithm examines, line verification only checks the local consistency based on the lines between two given points which might represent key-points, key-regions, bounding boxes or key-region pairs. However, the computational complexity of line verification is very expensive when one-to-many matching is allowed. Hence,we propose to perform line verification in two steps: 1) tentative bounding boxes searching according to the transform matrices that estimated based on the corresponding matches; 2)examine the spatial consistence between query and the matches from each bounding box.

The rest of the thesis are organized as follows: Chapter 1 will introduce the objectives, the challenges of our research and the contribution we made; Chapter 2 will discuss the state-of-the-art methods; a real-time retrieval scheme will be introduced in Chapter 3 to illustrate the benefit of integrating the structural information for document retrieval; in Chapter 4, a detection algorithm that is able to extract the multi-scale key-regions and the explicitly structure of document images will be presented; Chapter 5 will introduce a BoW-like method to efficiently embed the document structure into a histogram representation; a generic scheme to handle various of document retrieval scenarios will be finally described in Chapter 6 while the scheme to tune the similarity measurement between structural and visual features is also presented; Chapter 7 will introduce a flexible method to check the spatial consistency and compute the transformation between query and the its matches; the conclusion and future work will be discussed in Chapter 8.

Chapter 2

State of Art

Document retrieval problem has attracted many researches in the past decades and been tackled from different viewpoints such as content-based, structure-based, text-based signature and logo retrieval.

Optical Character Recognition (OCR) performs character analysis on scanned documents and translate the digital image into ASCII texts. During retrieval process, the matching score between documents is measured by the similarity on the corresponding texts. The OCR-based methods demonstrated promising advantage on processing document from the viewpoint of natural language since the synonym, cross-language etc. information could easily be taken into account. Unfortunately, the applications of OCR-based methods are retarded due to the recognition error and the matching complexity especially when the synonym and cross-language are considered. To reduce the degradation introduced by OCR errors, a post processing scheme is performed in [1] by a clustering process that locate the correct target term for the misspelled text. Besides, in [2, 3], the error correction process is performed based on the probability models that employ the information of the term frequency. In [4], the problem has been tackled by training language models to recognize the mis-spelled terms. Another factor that mainly hinders the growth of OCR-based methods is the matching complexity among the ASCII texts given the scenario that the amount of texts in the database are increasing significantly. In [5], the problem is approached by approximate string matching technique through text compression methods. Many other approximate string matching algorithms are surveyed in [6].

Another viewpoint to retrieve document from database is by matching the given symbols such as logos or signatures etc. For example, the logo-like regions are extracted by the corresponding information on their size, shape, compactness and location in [7]. The logo candidate regions are then further recognized by their proposed geometric invariant features. In [8], a system to recognize if the given logos exist in the incoming document is proposed based on local key-points matching while in [9] they further locate the logo position by sliding windows techniques over Blurred Shape Models (BSM) [10] description. Very similarly, in [11], the local key-point features are employed while the matches is further refined by a clustering process that proposed in [12]. Afterward, they further refine the matches by spatial consistency check process through homography estimation in [13]. In [14], the logo retrieval problem is tackled through matching the local key-point features followed by a geometric consistency validation process. Besides, the logo retrieval problem has also been concerned for natural scene images in [15] through several local features such as shape context [16] and sketches [17].

On the other hand, signature is another type of symbol that usually employed to retrieve documents from the database. A signature segmentation method is introduced in [18] through classifying the connected components into printed text and signature by local feature description extracted from each component. Instead, in [19], the text regions are removed by applying Bayesian approach on the two-dimentional features: aspect ratio of bounding box and the normalized contour size. Afterwards, the author employ Gradient, Structural and Concavity (GSC) [20] binary feature to describe the signature regions and employed to match one signature to another. In [21], a system is exploited for document retrieval based on signature matching while the signature regions are described by their proposed scale and rotation feature. Differently with the local features, Zhu et al. [22] proposed a novel multi-scale approach that improved the signature regions segmentation performance by capturing the structural saliency and dynamic curvature of 2D contour fragments. The authors further proposed two novel measures of shape similarity based on anisotropic scaling and registration residual error for signature matching.

The signature-based and logo-based follows similar step: specific region segmentation and feature extraction. At the end, the similarity score is measured based on the features extracted from the logo or signature regions. Even though such symbolic regions based method achieve many attention and successfully applied in various businesses, it is difficult to generalize those methods to other documents retrieval scenarios that do not seek logo or signature such as journal paper, books etc. Consequently, for the situation that aiming at solving more generic document retrieval problems, it is advisable to focus on more general characteristics of the document images such as the structure and content.

2.1 Layout Analysis based Retrieval

Layout is the most straightforward and explicit form of document structure and hence well studied in the past decades. Layout analysis methods explicitly segment document images into regions or blocks with logical or physical labels. Then a process comparing the structural relation among the blocks is performed to obtain the similarity between query and target images [23]. Generally, document layout analysis algorithm is implemented into two steps: document segmentation and blocks matching.

2.1.1 Document Segmentation

Document segmentation problem are researched for many situations whereas layout analysis is the most common one. In the past decades, the segmentation have been tackled by various algorithms that could be generally be grouped into *top-down* and *bottom-up*. The *top-down* approaches split document into blocks which are classified and then further division are performed adaptively into text lines, paragraphs. For example, in[24, 25], the document images are segmented into text and graphics blocks through iterative run-length smoothing algorithm. Instead, segmentation problems are also tackled with projection profiles [26, 27]. However, those *top-down* methods assume that the blocks to be only non-skewed rectangular while, on contrast, the *bottom-up* methods are generally more flexible.

Bottom-up methods usually starts from the document element at lower level such as pixels [28] or connected components [29, 30, 31] or simple grid patches [32] and then grouped to higher level representation of the documents (e.g. words, text lines, paragraphs). For example, in [29], connect component is extracted from the document images and described by both the shape and surrounding context. Besides, the patches generated by evenly grid cutting might be alternatively employed as the document elements at starting level. For instance, [32] simply divides document images into small patches and extracts GLCM (Grey Level Co-occurrence Matrix) features [33](the Energy, Entropy, Sum Entropy, Difference Entropy and Standard Deviation of the patch) on each patch.

For *Bottom-up* segmentation methods, another concern is the classification of the fundamental elements (pixels, connected components or patches) and the grouping strategy. For instance, all the connected component are classified into either graphics, text or space through k-means clustering [34] based on the extracted features in [32]. Afterwards, the blocks at paragraph level are obtained by heuristic grouping based on the labeled connected component. In [29], the connect components are classified by Multi-Layer Perception (MLP) classifier [35] and the labeled components are grouped by nearest neighbor analysis which basically combines the surrounding component together if they have the same label. Besides, we should note that there are some methods that simply perform the well-designed morphological operation on binary images and generate the segmented blocks [36, 37].

2.1.2 Block Matching

After the documents are reasonably segmented into blocks, the next challenging step for layout-based document image retrieval is matching groups of blocks between query and database images. Unfortunately, measuring the similarity between one group of blocks and another is usually expensive. For instance, the document layout is represented as the XY-tree in [38, 39] while the tree edit distance is employed to measure the similarity [40]. The tree edit distance measurement method is improved by adding the tree grammar information in [41, 42]. Besides, in [43], the document is OCRed and segmented and thus generate a full-connected labeled graph for each document image. They manage to low down the complexity for approximately computing the distance between graphs that implemented in two consecutive steps.

The above discussed tree or graph matching methods are precise since the label of the blocks (nodes) and edges between blocks could be easily taken into account. However, such matching strategies are usually very time-consuming for calculation. Hence, the label of blocks and edges are discarded in [44]. Instead, overlap area ratio is employed to find the best correspondences between query and database blocks while the overall distance is computed as the weighted summation of every corresponding blocks matching error. On contrast, three algorithms (assignment, minimum weight edge cover and Earth Mover's distance) aiming at the block mapping problem are compared in [45] where the match score between two block are measured by either Manhattan distance of corner points or area overlap ratio of the blocks.

Layout-based methods are widely employed for full page classification and retrieval where the document structure matters for the application. However, such layout-based descriptors present several drawbacks for solving the retrieval problem in generic manner. On one hand, such methods are highly dependent to the performance of the block segmentation algorithm which might be unstable over various type of document images. On the other hand, computing the similarity between two groups of segmented blocks (normally represented as graphs) requires a computationally expensive mapping process that hinders the scalability of the final retrieval application. Besides, since the blocks that consist the document layout usually are extracted at paragraph level (rows or words level as well sometime but ultra expensive for block matching), the contents of the document which are essential for exact matching scenario hardly play important roles in the retrieval process.

2.2 Local Content Feature based Retrieval

Content is the most essential and valuable resource of the document images. In many cases, OCR process is applied on the document images after digitalization. However, as discuss above, the OCR-based information retrieval methods are limited due to the recognition errors and the problems of text matching such as string edit distance computation, multi-lingual, synonym etc. Hence, plenty of efforts have been alternatively made for document image analysis aiming at content feature extraction that interpret the local regions (related to ASCII texts) into feature vectors.

2.2.1 Local Representation

To efficiently extract the main information from digitalized images, numerous efforts have been made to effectively extract features from various types of images such as natural scene, medical, document images etc. Afterwards, a group of local feature vectors is usually employed to represent each image and utilized for image classification, object detection, image retrieval etc. The local feature exaction process can further divided to two steps: feature detection and feature description.

Feature detection algorithms are applied to return the representative interesting objects of images such as Canny edges [46, 47], Harris corners [48], blobs [49, 50] and regions [51]. Among those detection algorithms, the most popular ones are Difference of Gaussian that applied in Scale Invariant Feature Transform (SIFT) [49], Determinant of Hessian that employed in Speed-Up Robust Feature (SURF) [50] and the Maximal Stable Extremal Regions (MSER) [51]. After the interesting objects are successfully detected, the next step for feature extraction is to describe the objects into vectors. In the past decades, plenty algorithms have been developed for describing the objects from different viewpoints such as contour (Fourier [52], Wavelet [53], Curvature Scale Space [54]), content(SIFT,SURF,OpponentSIFT [55], Histogram of

Oriented Gradient(HOG) [56]) etc. At the end, each image is represented as a group of feature vectors extracted by the description algorithms.

Such feature extraction strategies have been successfully applied to various scenarios in computer vision domain such as panorama stitching [57, 58, 59], object recognition [60, 61, 62], 3D scene modeling [63, 64, 65], natural scene image understanding and image retrieval [66, 67] and also introduced into document retrieval scenario. For example, document images are described with SIFT in [68, 69], SURF in [70, 14] and with HOG feature vectors in [71]. The most promising advantage for representing each image as a group of feature vectors, in our case, is that it is feasible to match image part with a full page images. Besides, such local content based features (SIFT, HOG, etc.) are generally very discriminative and thus obtain good performance when the exact matches are expected.

Even though many key-point/key-region detection algorithms have been applied to document images, the extracted interesting objects usually are not really semantical. For example, SIFT algorithm only returns corners, edges and the spaces between texts while MSER method roughly equivalent to Connect Component Analysis whereas most of the extracted key-regions are letters for binary document. Hence, based on the fact that characters are placed closer to each other than words are which are in turn placed closer to each other than paragraphs or columns are, we proposed Distance Transform based MSER methods to extract multi-scale/multi-level key-region such as letters, words, paragraphs. We should point out that besides the multi-scale key-regions, DTMSER algorithm can extract document structure as a dendrogram about how one key-region merge to another. This work has been published in [72] and will be explained with further details in Chapter 4.

It is very effective to represent document images as groups of feature vectors for the document retrieval applications. However, computing the similarity of a group of feature vectors with another group is usually not efficient comparing with representing each image as one single vector (e.g. BoW), especially when the target images are full-page images and consist of plenty of local features.

2.2.2 Global Representation

Various methods have been proposed to globally represent each document as a statistical vector based on local content features. For example, Li et al. [73] represent each document image as a sequence of word sizes in terms of the number of object pixels. Meng et al. [74] represent document images as the vertical and horizontal projections of both object pixel and crossing number(the number of changes from object to background and from background to object).

Besides, the local key-regions/key-points based description methods that widely employed in many scenarios of computer vision domain are also successfully applied to document image retrieval applications [69, 75]. It is would desirable to represent each image as one single vector based on those discriminative local features. Feifei Li et al. proposed Bag-of-Words framework to represent each image as a histogram vector over local features for natual scen image categorization [76]. In [77], document images are represented as Bag-of-Words histograms based on SIFT description. Similarly, in [78], regions are segmented from document images (see [79, 80]) and described by both HSV color histogram [81] and Local Binary Pattern (LBP) [82]. At the end, Bag-of-Regions methods is proposed to globally represent document images. Globally describing document as one feature vector per image could encode the discriminative local features and is efficient for classification or retrieval at full page level.

2.2.3 Encoding Structure for Local Features

As discussed before, document structure is very valuable for searching similar images from database. However, the layout-based methods that explicitly express the document structure are not desirable in many cases due to the segmentation errors and matching complexity. Hence, implicitly encoding the spatial information of the local content is alternatively employed to represent the document structure. For instance, in [83], the document image is iteratively divided into finer parts and the average of pixel intensity of all the resulted parts is concatenated and employed to describe the document, while in [84] the Run Length(RL) encoding, and in [85] SIFT, are used to describe the resulted document parts. A problem of such pyramidal spatial method to encode structure information is the exponential increase of the dimension of the final representation vectors. Hence in [86, 87], the images are recursively partitioned into halves, instead of pyramid, and thus leading to lower dimensionality. As another solution, each part of spatial pyramidal BoW representation is analyzed according to its origin windows in [88] and only the windows that show higher discriminative power are used in the final representation.

However, for document image retrieval, such spatial information based method do not explicitly encode low-level structural information, but rather the spatial distribution of local patterns. Hence, we proposed to employ the key-region pairs, instead of single key-region or key-point, as the elementary pooling unit for BoW representation. In such pair-wise way, we manage to embed the explicit document structure (*inclusion* relations between key-regions) into BoW framework. This work has been published in [89] and will explained in details in Chapter 5.

Adding the structural information (either implicitly or explicitly) into a global representation of documents might significantly improve the efficiency of full-page retrieval scenario, but it is not suitable for searching image part from database. As we discussed before, our research aims at generic framework for both full-page document and just part document retrieval. Hence, it would be better to represent documents as groups of feature vectors that contain the content information of document. The most challenging point here is to describe the structural relations among such local features since our generic framework aims to tackle the whole spectrum of problems from exact, content-based, matching to purely structure based matching.

DTMSER [72] algorithm provides an efficient way to extract explicit document structure as a dendrogram (hierarchical tree) whose nodes are related to multi-scale key-regions and branches correspond to *inclusion* structural relations. For example, a letter-level key-region is included by another key-region which corresponds to the word that it belongs to. Comparing with the global spatial information that widely employed to express the image structure, the dendrogram conveys rich source of explicit structural information such as local *inclusion* relation and *left/top of* between key-regions. The most attractive advantage of such structural description is that it is suitable for both part-based and full page queries through pair-wise key-regions querying that combines the explicit document structure and local content features.

To address the problem of complexity for computing similarity among content features, plenty of efficient algorithms such as Approximate Nearest Neighbors (ANN) [90], Locality Sensitive Hashing (LSH) [91], feature clustering combing with inverted file indexing [92], Product Quantization (PQ) [93], Bucket Distance Hashing (BDH) [94], k-d tree [95, 85] have been proposed or applied to document image retrieval domain. However, the indexing strategy on the spatial relations between features (e.g. *inclusion* relation and *left/top of*) have not been exploited in our domain.

Spatial databases [96] are special databases designed for dealing with the spatial relations (*inclusion*, *intersection*, *overlapping* etc.) among geometrical objects such as points, lines, polygons, etc. while typical databases are designed to manage various numeric and character types of data. Thanks to the advanced spatial indexing techniques, spatial databases allow to cast queries in terms of geometrical relationships among the stored objects in efficient fashion. For example such databases support queries such as "*retrieve all the objects having a border close to point **A** that overlap with circle **B** and intersect with the polygon **C***". They have been widely used in various Geographical Information System (GIS) applications such as maps, national census, car navigation, global climate change research, etc. however, to our best knowledge, they have not been exploited in the document analysis community up to now. The details about this will be further explained in Chapter 6.

Chapter 3

Real-time Pyramidal Document Structural retrieval

As discussed in the Chapter 1, our researched is focused on a generic framework where both structural and visual similarity is measured. Hence, in this chapter, we will initially explore how important the structural information of document images would be in the retrieval process. As discussed in Section 2.1, due to the block segmentation and matching problem, it is not proper to extract structural information as the layout of the documents. Instead, we employ a pyramidal decomposition feature [97] as the representation where the structural information is encoded as the rough locations of local patches. The main purpose of this chapter is to illustrate that the structural information of document images would make the simple features (pixel intensity) to be very discriminative in a real-time document retrieval scenario.

In this chapter, we will test the performance of such spatial pyramid structural feature for newspaper retrieval scenarios at full page level. Since the datasets are dramatically increasing everyday, the computation complexity on image description, feature similarity and retrieval should be particularly be considered. Hence, to extract feature vectors of newspaper images, we choose a global descriptor as the whole page similarity. Such strategy demonstrates low computation complexity as only one feature vector with fixed length is extracted for each image. On contrast, local descriptors extract many feature vectors (normally thousands or more) for each image and voting process should be additional applied. As our application aims at searching images in daily 'exploding' large dataset, a global descriptor with low computation is chosen here to extract feature vector.

3.1 Global Image Description

Pyramidal decomposition provides an effective and global wayto represent images the integrate structural and local information. It expresses the pixel intensity of image at different scales. The algorithm is performed by a recursive operation of cutting the images into four rectangular regions, the intensity values of which are used as feature vector. In practice, the number of iterative cuts represents the detail-capture ability

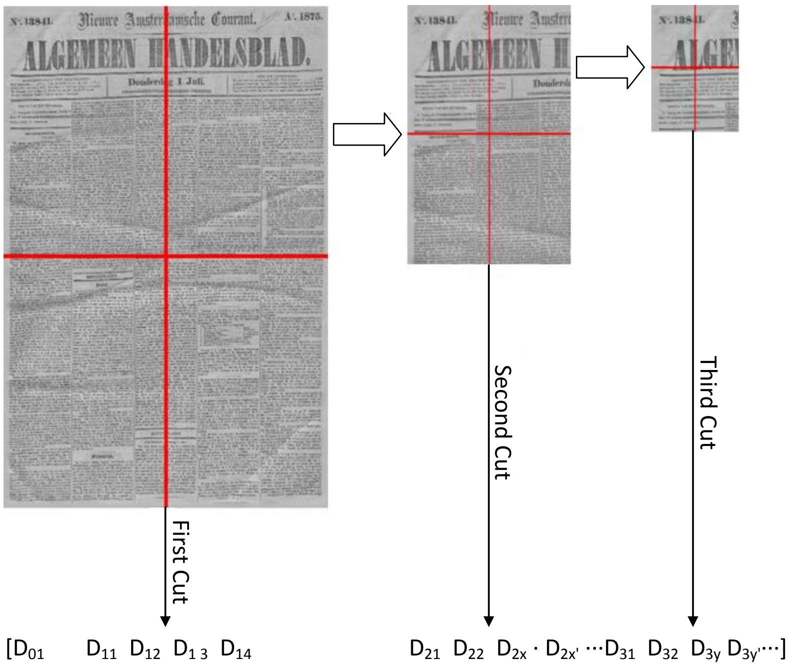

ALGEMEEN HANDELSBLAD.
First Cut
Second Cut
Third Cut
[D_{01} D_{11} D_{12} $D_{1\,3}$ D_{14} D_{21} D_{22} $D_{2x} \cdot D_{2x'}$ ···D_{31} D_{32} D_{3y} $D_{3y'}$···]

Figure 3.1: Demonstration of Pyramidal Decomposition. The extracted feature vector is showed at the bottom.

and defines the length of feature vector. In this chapter, we mainly test such feature extraction method in the scenario of retrieving the front-pages and non-front-pages for digitalized newspapers. As demonstrated in Figure 3.1, the first level (D_{01} in feature vector) corresponds to the intensity over the whole image, the second level gives the intensity of 4 rectangular cuts: D_{11}, D_{12}, D_{13} and D_{14}. Consequently, 5 level cut overall returns a feature vector with 341 (1+4+16+64+256) elements, and sixth level returns 1365 elements.

The pyramidal decomposition descriptor is scale invariant because the resolution or scale change does not lead to the pixel intensity alteration. Besides, the feature vector extracted by pyramidal decomposition are slightly tolerant to translation and rotation. The extent of such tolerance depends on the feature vector level: the higher level cut, the finer feature vector extracted and hence the more sensitive to the translation and rotation. In general, scanned document images exhibit limited skew, hence the pyramidal decomposition descriptor is an adequate choice.

3.2 Retrieval by Similarity

Several types of distance metrics are introduced to measure similarity for different applications. A.Vadivel [98] compared four different distance metrics (Manhattan, Euclidian, Cosine, and Histogram Intersection) with a colour histogram based descriptor and concluded that Manhattan distance metric is better than others for content-based image retrieval. However, Manesh Kokare [99] showed that the Canberra distance metric is the best one for their content-based image retrieval after compared various distance metrics, including Euclidean, Mahalanobis, Chi-square etc. for texture features extracted by Gabor filter bank. Obviously, the 'best' distance metric varies over different descriptors and hence it is necessary to test which distance metric is 'best' for our pyramidal decomposition descriptor (feature vector).

For retrieving correspondences among the daily added newspaper images, the similarity computation between the query and the existed dataset images is inevitable and has to be done on-line. Consequently, besides effectiveness, its complexity should be especially considered. In this chapter, four commonly used distance metrics are considered to measure the dissimilarity between the query feature vector and dataset images. We define the feature vector of query image as $\mathbf{Q} = (q_1, q_2, \cdots, q_n)$ and the dataset newspapers as $\mathbf{I} = (i_1, i_2, \cdots, i_n)$. Here, q_j and i_j represents the jth elements of feature vector of the query and dataset image respectively. The different distance metrics between $\mathbf{Q}$ and $\mathbf{I}$ are defined as follows.

- **Euclidean distance** is the most frequently used metric, easy to understand and usually effective in many cases, to calculate the distance between two vectors. We can calculate the Euclidian distance between $\mathbf{Q}$ and $\mathbf{I}$ as:

$$d(\mathbf{Q}, \mathbf{I}) = \sqrt{\sum_{j=1}^{N} (q_j - i_j)^2} \tag{3.1}$$

- **Chi-square distance** is a special type of Euclidean based distance which is calculated in a weighted way. Suppose $C = (c_1, c_2, \cdots, c_j, \cdots, c_n)$ denotes the average feature vector of the images in the dataset, then the chi-square distance between $\mathbf{Q}$ and $\mathbf{I}$ will be:

$$d(\mathbf{Q}, \mathbf{I}) = \sqrt{\sum_{j=1}^{N} (q_j - i_j)^2 / c_j} \tag{3.2}$$

- **Cosine distance** measures the difference between two feature vectors $\mathbf{Q}$ and $\mathbf{I}$ by calculating the vector cosine angle between them, and so its value ranges in [0 1]. One important property of the Cosine distance is that it just takes the included angle of two feature vectors into account while their scale difference is 'ignored'. The Cosine distance between $\mathbf{Q}$ and $\mathbf{I}$ is defined as:

$$d(\mathbf{Q}, \mathbf{I}) = 1 - \frac{\sum_{j=1}^{N} q_j * i_j}{\sum_{j=1}^{N} q_j^2 * \sum_{j=1}^{N} i_j^2} \tag{3.3}$$

- **Histogram intersection distance** computes dissimilarity between the query image $\mathbf{Q}$ and a stored image $\mathbf{I}$ in the database, using histogram intersect distance, is given by:

$$d(\mathbf{Q},\mathbf{I}) = 1 - \frac{\sum_{j=1}^{N} min(q_j, i_j)}{min(\sum_{j=1}^{N} q_j, \sum_{j=1}^{N} i_j)} \tag{3.4}$$

As distance metric computation is an iteratively progress for each image of dataset at on-line manner, computation efficiency is a major factor to make choice. Among the four introduced distance metrics, Cosine distance metric possesses the lowest computation complexity as it boils down to a simple multiplication of corresponding elements and a summation if the data has been normalized previously. Besides, for pyramidal decomposition feature vectors of the newspaper images, because the scale of feature vector is 'ignored' in Cosine distance metrics, such dissimilarity measure also hold the invariant to illumination change, which commonly happens for scanned images. So in Section 3.4.2, Cosine distance showed its advantage on both efficiency (fast to calculate) and effectiveness (ability to assess visual similarity correctly given the feature) over other distance metrics.

3.3 Relevance Feedback

Apart from the particular distance metric used to assess similarity, it is expected that the number of false positives for non-front-pages might arise in the retrieved results due to the large inner class variation. One way to solve this problem is to obtain feedback from the user about the retrieved result. Such relevance feedback would be helpful to make better understanding on what the user really wants, and then use such information to improve the performance in later retrieval iterations. There are two main methodologies to integrate user feedback to the process. These involve to re-design the query vector or re-rank the feature vectors of dataset images according to the user's feedback.

In details, the first step uses an initial query vector (corresponding to the query image) to retrieve several items. Afterwards, the user is asked to indicate which images are correct results within the first few items of the retrieved set. Subsequently, by using the provided relevant information, an adaptively revised query formulation or re-ranking process of the dataset images is performed hoping to retrieve more relevant items during the subsequent search.

Relevance feedback, which follows the above idea of using the relevant information from the user, is a controlled method for improving the performance of the retrieval system by interacting with the user about the truth in previous retrieved results. As showed in Figure 3.2, after an initial retrieval step, the user is asked to provide some feedback about which results are correct (the images with green frames are supposed to be pointed out by user). This relevance information is used by relevance feedback which allows to reformulate the query and hence provide an enhanced result list in subsequent iterations.

Here, we have tested two different relevance feedback methods. The Rocchio method, is a relevance feedback algorithm that follows the idea of query reformulation

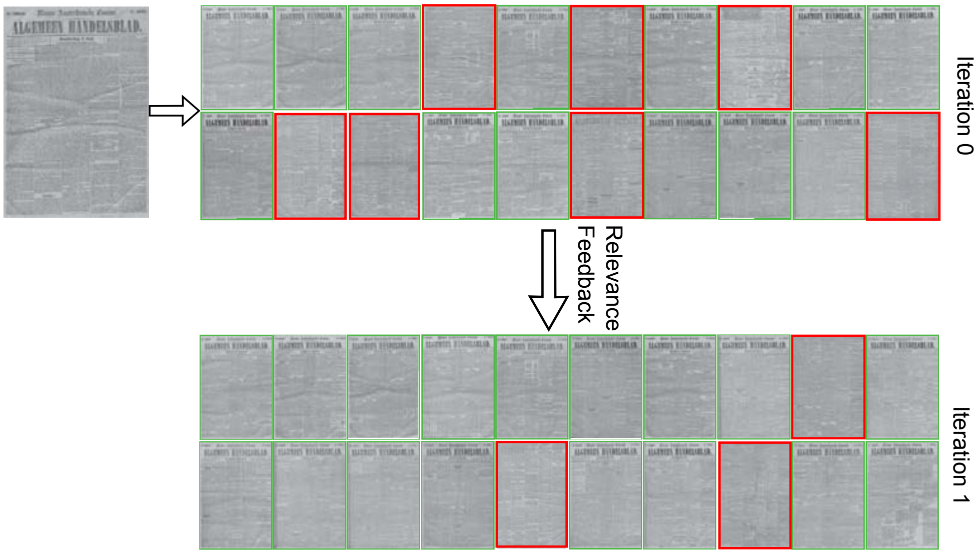


Figure 3.2: Demonstration of Relevance Feedback. The documents with green frames are the true relevant images while the images with red frames are falsely recognized as relevant. As shown here, with one iteration of relevance feedback, the MAP performance increases from 0.7257 to 0.9308

trying to find, given the relevance assessments, a new query point in the vector domain that is closer to the positive samples and farther from the negative ones than the original query point. The Relevance Score method is a re-ranking method, that tries to reorganize the original resulting list in terms of the relevance assessments without casting any new query.

For these two relevance feedback method, both of which need iteratively calculating similarity during a relevance feedback. However, processing the Euclidean distance applied to a large database can be really time consuming.On the other hand, the Cosine distance metric boils down to a simple vector multiplication if all the feature vectors have been normalized previously. Besides, it can be shown that, as far as the ranking of elements is the only concern, the Cosine distance metric serves equivalently as the Euclidean distance metric. Consequently, without loss of generality, a Cosine distance based relevance feedback method is employed here for evaluating different relevance feedback methodologies, in order to improve the calculation efficiency.

3.3.1 Rocchio's Algorithm

The Rocchio's algorithm [100] is a widely used relevance feedback strategy. At each iteration about the so far retrieved result, the Rocchio's algorithm makes use of user's feedback to reformulate the query in order to incorporate the relevance feedback information into the vector space model. Taking the mth feedback iteration for instance, the query vector reformulation $\mathbf{q}_m$ is computed as

$$\mathbf{q}_m = \alpha \mathbf{q}_{m-1} + \frac{\beta}{|D_r|} \sum_{\mathbf{d}_j \in D_r} \mathbf{d}_j - \frac{\gamma}{|D_n|} \sum_{\mathbf{d}_j \in D_n} \mathbf{d}_j \tag{3.5}$$

where $\mathbf{q}_{m-1}$ is the reformulated query vector in previous iteration of relevance feedback , and D_r and D_n the sets of relevant and non-relevant documents affirmed by the user respectively. α, β and γ are the associated weights that reformulate query vector with respect to the query used in previous iteration of relevance feedback, the relevant and non-relevant items. In our setup, we experimentally set the weighting values to $\alpha = \beta = \gamma = 1$ which result in the equal weight for previous query, average positive and negative samples.

By emphasizing the visually similar features contained in relevant images and removing the visually different ones that wrongly retrieved, Rocchio's algorithm could gradually move(revise) the query closer to the relevant ones, and farther away from the irrelevant ones, in the feature space. In another words, Rocchio's relevance feedback algorithm leads to smaller inner-class and bigger intra-class distances. Hence, retrieving by the revised query would lead to better performance.

3.3.2 Relevance Score

Another method of relevance feedback is to revise the similarity measurement according to relevant information obtained from the user. Relevance Score method proposed by Giorgio Giacinto [101] provides a way to adaptively revise the similarity measurement. The idea of this method is to define similarity between query and each document in the dataset as the ratio between the nearest relevant and the nearest non-relevant document images. The relevance score RS is computed as follows:

$$RS(\mathbf{I}) = \left(1 + \frac{dR(\mathbf{I})}{dN(\mathbf{I})}\right)^{-1} \tag{3.6}$$

where $\mathbf{I}$ represents the image in the dataset, $dR(\mathbf{I})$ represents the distance between the image I and the nearest relevant image, and $dN(\mathbf{I})$ represents the distance between query and the nearest irrelevant image retrieved so far.

The advantage of this method is that it is capable to 'remember' the previous retrieval result: for the image $\mathbf{I}$ retrieved to be true front-page, $dR(\mathbf{I}) = 0$ and so $Relevance(\mathbf{I}) = 1$ which means image $\mathbf{I}$ is exact visually similar to the query in this iteration. Analogously, for the non-relevant image $\mathbf{I}$, $Relevance(\mathbf{I}) = 0$ which means image $\mathbf{I}$ is totally different to the query. Consequently, the images marked as relevant will always be definitely similar and the non-relevant ones will be extremely different with the query in the later iterations. On contrast, Ricchio's algorithm needs to compute the distance between the new query and the dataset images.

3.4 Experimental Results

3.4.1 Dataset and Evaluation Measures

The experimental setup is implemented in two steps: we first extracted the feature vectors of all images in the database in an offline fashion. Subsequently, we evaluated the performance of four alternative distance metrics discussed in section 3.2 based on a subset of 500 newspaper images comprising 2 classes (front-page and non-front-page from a single newspaper title). This experiment aimed to test the effectiveness of the feature vector extraction process and determine the best distance metric for newspaper retrieval. We execute the two previously stated relevance feedback strategies using the pyramidal decomposition feature vectors and the tested distance metric over the whole newspaper dataset containing 23004 images comprising 16 classes (front-page or non-front-page for 8 different newspaper titles). In each of the experiments, we use the Mean Average Precision (MAP) to evaluate the performance of the system.

3.4.2 Evaluation of different distance metrics

In order to evaluate the performance of the different distance metrics, we performed an experiment using a subset of the newspaper dataset consisting of 500 images of a single title which were previously classified in two classes (108 front-pages and 392 non-front-pages).

In addition, to study the effect of adding more detail in the pyramidal decomposition feature, we repeated the experiment using different levels of decomposition.

We compute the precision-recall curve and MAP for each image in a leave-one-out fashion. Each image is taken as query and we perform the retrieval versus the remaining 499 images. Consequently we obtain 500 precision-recall curves and 500 MAP values for each distance metric.

In order to evaluate the performance of different distance metrics based on the same pyramidal decomposition feature, we calculate the average precision-recall curve and MAP value according to the 500 precision-recall curves and MAPs for each type of distance metric (see Figure 3.3 and Table 3.1). L4, L5, and L6 in Table 3.1 and Table 3.2 represent the feature vector level we used, and Euc., χ^2, Cos., HI correspond to Euclidian, Chi-Square, Cosine, Histogram Intersection distance metric respectively.

Table 3.1: Average MAP of all the queries.

		Euc.	χ^2	Cos.	HI
All images	L4	0.8247	0.8266	**0.9438**	0.7337
	L5	0.8364	0.8390	**0.9342**	0.7318
	L6	0.8615	0.8652	**0.9326**	0.7384
Only front-page	L4	0.5820	0.5843	**0.8939**	0.4279
	L5	0.6555	0.6585	**0.9255**	0.4580
	L6	0.7469	0.7508	**0.9461**	0.4977
Only non-front-page	L4	0.8916	0.8933	**0.9575**	0.8180
	L5	0.8863	0.8887	**0.9438**	0.8072
	L6	0.8931	0.8967	**0.9289**	0.8048

We also perform the experiment on the front-page subset (only the front-pages are used as queries) and non-front-page subset (only non-front-pages are used as queries) separately in order to evaluate the performance on certain classes of images (see

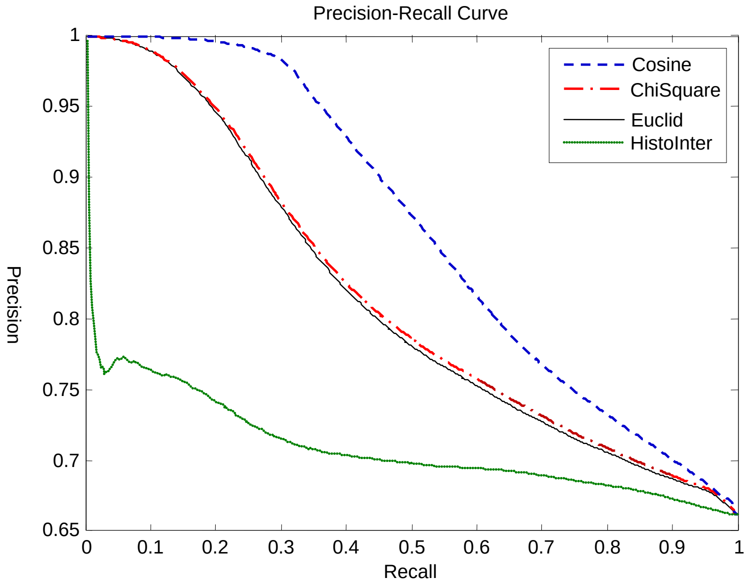


Figure 3.3: Precision-Recall curve of 500 queries (both front-Pages and non-front-Pages, L5 feature vector).

Table3.1).

Despite the different level of feature vector extraction and types of dataset, the Cosine distance performs much better than any other distance metric consistently because such similarity measurement is invariant to the illuminant change that commonly exists in the grayscale documents. Euclidian distance and Chi-Square distance yield similar performance with the latter being marginally better, while Histogram Intersection distance invariably yields the poorest performance during the whole experiment. Consequently, we arrive to the conclusion that the Cosine Distance is much more suitable for newspaper image retrieval scenario.

For front-page only images retrieval, increasing the level of pyramid decomposition feature results in clear effect: it produces better performance no matter what the distance metrics are used. However, for non-front-pages only retrieval, it does not yield any better performance for higher level feature. This is to be expected because all the front-pages contain a pretty fixed part (the title part) which would play more important role at higher detailed levels. The non-front-pages on the other hand are similar only at the level of the general structure (e.g. all will have the same number of columns) which is already captured at low pyramid levels. Consequently, adding more detail by extracting higher level feature actually leads to worse performance.

We also calculated the computational cost of extracting the pyramidal decomposition feature over the 500 query images ('Descriptor' in Table 3.2) as well as the distance calculation between the query image and other 499 dataset images ('Distance' in Table 3.2). The time shown in Table 3.2 is the average time consumption over 500 queries. As demonstrated in Table 3.2, the computational complexity will rise up sharply for both feature vector extraction and distance computation when increasing the pyramid level. Besides, we should note that even though distance calculation take much less time than feature extraction for a dataset consisted of 500 image, the time consumption on distance calculation will increase linearly as the dataset size grows while the online computation of feature extraction is independent.

Table 3.2: Time Consumption of Feature Extraction and Distance Calculation.

Feature Level	Descriptor (s.)	Distance (s.)
L4	0.0277	3.3280e-05
L5	0.0445	9.2561e-05
L6	0.0979	3.5360e-04

Besides, we also test the document representation method on an administrative dataset consisting of 4109 binary invoice documents. Following the previous experimental settings, the experiments are repeated for 4 different similarity measures while leave-out strategy is employed to generate the queries. Whether one image is relevant with another is determined by checking if the two invoice are provided by the company assuming they share the same template and thus similar format and layout.

As shown in Table 3.3, for the binary invoice images, the Euclidean and Cosine distance generally performs promising better than the other two similarity measurement. The performance difference between Euclidean and Cosine distances is not notable here. It is because, theoretically, the two distances show high correlation when the scales of query and database images hold small variation ($\|\mathbf{Q}\| \simeq \|\mathbf{I}\|$). It is proved in Appendix B. On the other hand, since all the images in this experiment are binary, the scales of their representation vectors hold less variation than the ones at grayscale level do.

Since all the invoice images are in black-white format, thus the pixel intensity feature encoded in the image representation is less discriminative as it does for grayscale image in the previous experiment. Hence, we also performed our experiment at higher pyramidal level (Level 7) to further enhance overall discriminative power of the employed representation. For the Euclidean similarity measurement, it is illustrated that the feature representation at level 7 achieved the best performance with 1 percent improvement over level 6. Besides, approximately equivalent behavior with Euclidean distance is observed for Cosine measurement. Even though it might lead to slighter better performance, we stop from going upper pyramidal level because the feature extraction process will be not real-time anymore (nearly 2 seconds each image at level 8).

3.4.3 Results on Relevance Feedback

Although the retrieval performance is good on small datasets as demonstrated above, retrieval performance decreases when generalizing the previous retrieval method to the whole newspaper containing 23004 images comprising of 16 classes. Consequently, we perform relevance feedback strategies on the whole newspaper dataset to improve the user's retrieval experience by offering more precise query vector or similarity mea-

Table 3.3: Average MAP of all the queries.

	Euc.	χ^2	Cos.	HI
L4	0.8666	**0.8687**	0.8667	0.4283
L5	**0.9170**	0.9061	0.9147	0.6412
L6	**0.9487**	0.8972	0.9426	0.7024
L7	**0.9568**	0.6266	0.9504	0.6841

surement. Figure 3.2 demonstrates an example on how Rocchio's relevance feedback method improves the retrieval result. One image is randomly picked from the newspaper database as query, and then we search for the relevant images in the rest of the dataset using cosine distance metric. Subsequently, the information about which images in the initial result (Iteration 0) is really relevant is obtained from the user (GroundTruth labels in our simulated experiment here). Such information is then used by the relevance feedback method to improve the retrieval result (Iteration 1). As we can see in Figure 3.2, only 13 out of top-20 retrieved results are relevant in the initial retrieval attempt and the corresponding MAP value is small (0.7257). However, during a single iteration of relevance feedback, 17 out of top-20 are found and the MAP value increases (0.9308).

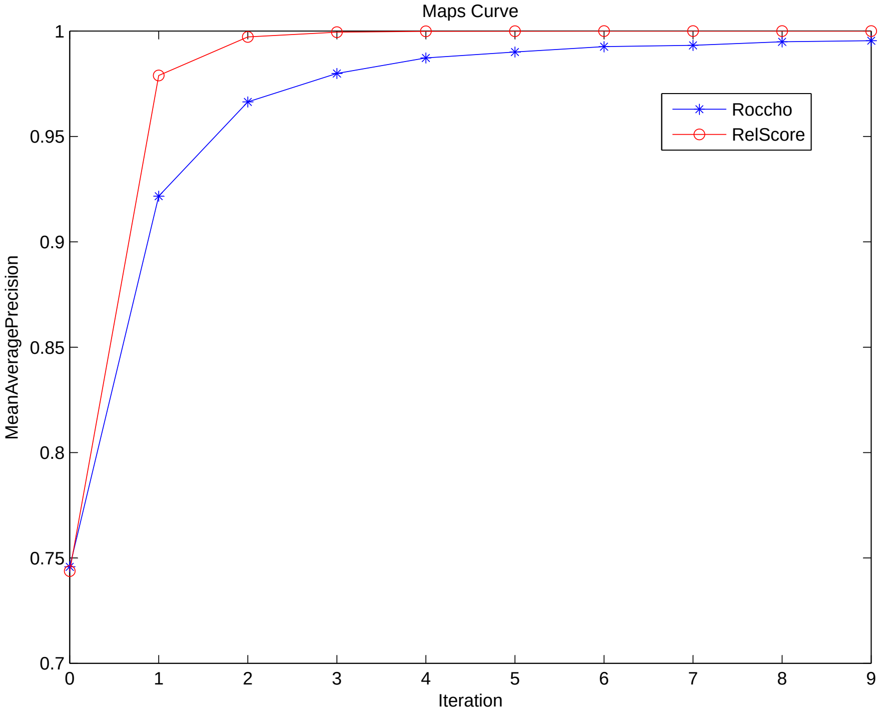


Figure 3.4: MAP improvement by Relevance Feedback.

Within the 23704 images, 2000 images are randomly chosen as queries. For each query, 100 relevant images are expected to be retrieved and the relevance feedback step is executed no more than 9 times over top-20 retrieved result (in a real application, the user could execute this step many times until the retrieved image satisfy the user). The feature vectors of all the images (both query and dataset images) are normalized before relevance feedback facilitating Cosine distance calculation.

The experiment was implemented as a simulation where the relevance feedback process is fulfilled as follows. We first labeled all the samples in our dataset. Then for each relevance feedback iteration the samples that bear the same label as the query are automatically selected as the relevant ones, in the same manner as a user would do.

For each query we record the MAP value at each iteration during the relevance feedback process. At the end the average MAP value over the 2000 queries is calculated at each iteration. The result is shown in Figure 3.4. Rocchio and RelScore, the symbol in Figure 3.4, correspond to Rocchio's method and Cosine distance metrics based Relevance Score method.

As expected, as the Relevance Score Method variants are capable to 'remember' the relevant images retrieved so far (labeled so in previous iterations), they achieve much improvement than Rocchio's method. When applying Relevance Score Method, for most queries, after 3 iterations, all of the 100 top ranked images are relevant to the corresponding queries. However, when executing Rocchio's vector revision Method, the MAP performance converges to 100% much slower.

3.5 Summary

In this chapter we have introduced a retrieval-based application based on the visual (pixel intensity) features with the spatial pyramid structure of the images. We demonstrated that with the help of structural information, the simple intensity features that is very cheap to calculate achieved promising performance over a large dataset. Besides, Cosine distance is observed to be very efficient and effective for newspaper retrieval scenario. At the end, we demonstrated relevance feedback strategies could significantly improve the retrieval performance by interacting with the users. Relevance Score method which can 'memorize' the users choice showed its advanced capability over Rocchio's query revision strategy.

However, the introduced manner to encode structural information is not invariant to the rotation while it is increasing necessary to integrate such feature into the system in the scenario of camera based document retrieval. Besides, representing each document as a global vector is not advisable when the expected matches are specific portions of the images. Hence, in next chapter, we will introduce an alternative method to extract document structure to avoid the above drawbacks.

Chapter 4

Distance Transform based MSER

The document structure plays important role in document retrieval. As demonstrated in the previous chapter, benefiting from the discriminative power of the pyramidal document structure, the image representation based on the simple density feature achieves remarkable performance on both fast computation and precise result (could be further improved through a relevance feedback scheme). However, such an implicit pyramidal structure implies obvious drawbacks in part-based retrieval scenarios such as logo searching, address block matching and shopping item retrieving etc. In this chapter, we will exploit an efficient detector for document images to extract semantic multi-scale key-regions that roughly correspond to letters, words and paragraphs. We will demonstrate that such semantic key-regions slightly outperform SIFT and MSER detectors while they are more descriptive, requiring descriptors of smaller size when used in a bag of words framework . More importantly, the detection algorithm presented here could simultaneously extract document structure as a dendrogram of those multi-scale key-regions. The usage of such a structure will be discussed in the next chapter while this chapter will solely focus on the quality of the key-regions themselves.

Key-point or key-region based methods have achieved great success in various of applications and also widely employed for document analysis. However, as, it is difficult to fully associate meta-data to the images due to the exploding growth of document imagery data and, on the other hand, limitation on computation and storage. Consequently, the need for extracting region/points of interest has resulted to a plethora of recent advancements with the emphasis being on reducing the size of image descriptors without compromising retrieval efficiency [102].

Key-point correspondence based algorithms have also been used for image retrieval, either through a bag-of-words framework to create global image descriptors, or by direct key-point indexing in cases when part-based retrieval is significant. Such approaches are based on a variety of key-point and key-region detectors (e.g. Harris corner [48], Harris-Laplace and Hessian-Laplace [103], Difference of Gaussians [104], Hessian determinant [105], MSER [51], etc.) and an even larger number of local descriptors (e.g. SIFT [49], GLOH [106], SURF [50], HoG [56], etc.).

Although in the document analysis domain there is a chronic lack of large public datasets, the issue of retrieval in big collections of documents has always been a topic of interest with clear socio-economic impact especially in the administrative and the historical document analysis areas. Following suite from the domain of natural images, state-of-the-art key-point detectors and local descriptors have been successfully used in document analysis for document representation in classification and retrieval scenarios [107], as well as other applications such as logo spotting [8], etc.

The basic premise of key-point detectors such as SIFT and SURF is to detect as many stable key-points as possible in order to "densely cover the image over the full range of scales and locations" [108]. Although this makes a lot of sense for object recognition in individual cluttered scene images, it is not necessarily optimal for retrieval applications. Indexing large numbers of local features extracted from an equally large number of images is inefficient, even though it can become tractable through the learning of small codebooks [109] and state of the art hashing and searching techniques [110][94]. At the same time, document images are distinctly different to natural scenes as documents have an explicit structure and are generally high contrast images (giving rise to numerous stable key-points). Classically detected key-points, although they work reasonably well since they densely cover the document image, do not carry any particular semantic or structural meaning.

On the other hand, methodologies specifically designed for document images, make explicit use of document characteristics in their representations. As an example the document matching approach of Nakai et al. [111], makes use of structural features of the document and local topological information. In the case of [111] the centres of blobs detected through blurring and subsequent thresholding, assumed to correspond to words, comprise the key-points, while an affine invariant descriptor encoding the relative position of such blobs in their neighbourhood of the key-point is subsequently used. The indexing and retrieval scheme employed is extremely fast, able to retrieve at 40ms in a dataset of 10 million pages [112].

The approach of Nakai et al. [111] is indeed a very efficient solution given that the objective is exact document retrieval. More often than not though, what is of interest is to cover a wider range of retrieval scenarios rather than exact matching only. For example, similar documents might share whole paragraphs of text -in which case word blobs and a feature based on the relative positioning of words could provide a good basis for similarity search- but frequently, similarity is evident in the document structure but not in the exact content. See for example the documents in Figure 4.1. In this chapter, to evaluate the quality of the extracted key-regions that might be employed in different retrieval scenarios, we will focus on retrieving the invoices that are generated by the same provider which might not be similar in terms of their textual content, but still look visually similar.

In this chapter, we present the first steps towards such a document representation. We focus on the efficient detection of semantic key-regions that encode structural information among different levels (letters, words, paragraphs and so on). We demonstrate that the proposed key-region detector is efficient to calculate and results to a smaller number of semantic key-regions than other state-of-the-art key-point and key-region detectors such as SIFT and MSER. To demonstrate the advantage of the proposed key-region detector for document retrieval we calculate SIFT descriptors over the detected key-regions and use them for indexing and retrieval in an adminis-

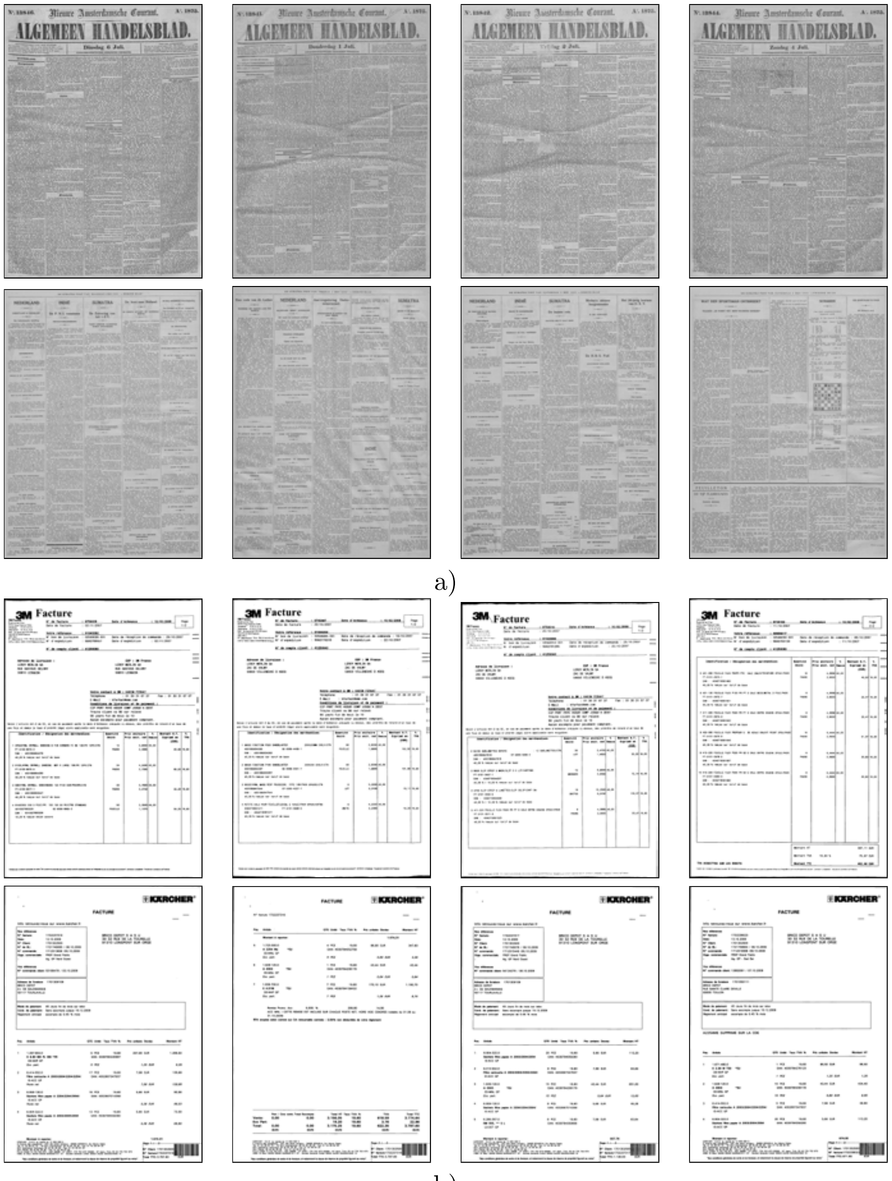


Figure 4.1: Typical applications of document retrieval include historical and administrative document analysis uses. (a) images taken from the IMPACT historical newspapers dataset, where a typical application is the retrieval of front pages - used with permission, (b) images from a typical digital mailroom page flow, a typical application being the retrieval of invoices from the same provider.

trative document scenario. We show that the key-regions detected with the proposed method yields better results than other state of the art key-point and key-region detectors.

4.1 Key-Region Detectors in Document Analysis

The standard local feature extraction pipeline that consists of a key-point detector followed by a local descriptor has yielded high performance in numerous challenging problems such as object recognition, robotic mapping and navigation, image stitching, 3D modelling, natural scene understanding, etc. Various detectors (e.g. Harris, Hessian, SIFT, MSER) and descriptors (SURF, HOG, SIFT) have been proposed during the past decades. In this chapter, we will concentrate on the performance of three different detectors (SIFT, MSER and our proposed DTMSER) when applied in the document image domain. The SIFT descriptor will be invariably employed to extract local features for all of the three considered detectors.

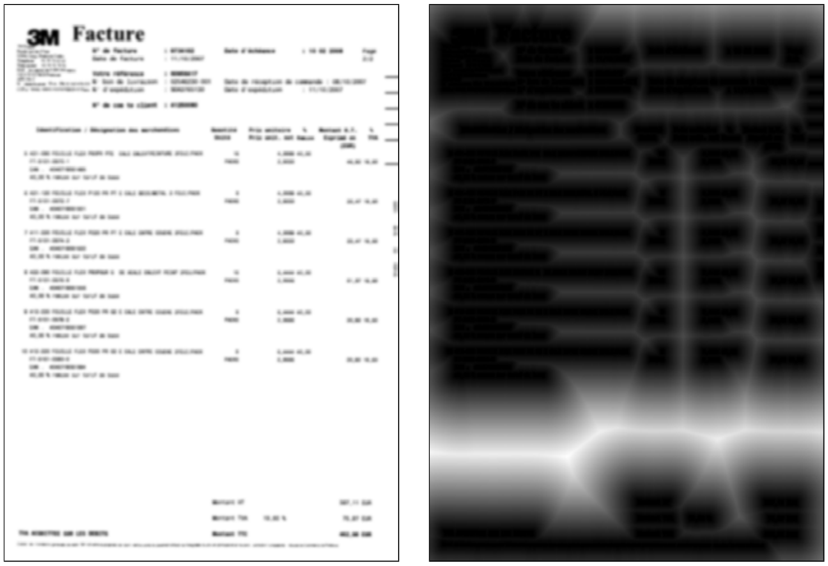
In the SIFT framework, key-points are defined as maxima and minima of the Difference of Gaussians (DoG) function applied in scale space to a series of smoothed and resampled images. It therefore detects salient and meaningful blobs at their best representative scale. However, when used in (usually binary) document images, the extrema of the DoG function is usually found at the lower scales, provoking that most of the extracted keypoints correspond to character corners, edges and spaces between characters, instead of higher-level entities. Such key-points are very stable, but present relatively low discriminatory power from the semantic viewpoint.

Concerning MSER, key-regions are extracted in terms of the stability of the intensity function over their outer boundary. As such, the algorithm detects blobs that present an important intensity change to their immediate surroundings. When used with document images, the set of maximal regions generally correspond to text parts (usually individual characters) and other dark foreground regions and the set of minimal regions to their white background counterparts. In the extreme case of bi-level images, the output of MSER is roughly equivalent to a connected component analysis. In the document analysis domain, MSER regions have been shown to perform well in matching tasks when dealing with "graphical" documents such as manga [113] comics.

4.2 Distance Transform based MSER

In the domain of document analysis, it is desirable to identify key-regions that relate to the structural elements of the document, namely characters, words, lines and paragraphs, as they carry important semantic information . Moreover, this should be done in an efficient, repeatable and stable way, as opposed to existing layout analysis approaches which are generally exhaustive and inherently unstable.

The notion of scale in the case of documents is tightly linked to the distance between the structural elements of the document. Characters are placed closer to each other than words are, which are in turn placed closer to each other than paragraphs or columns are. Moreover, the hierarchy of these structures is well defined and informative. On the other hand, the MSER algorithm provides an efficient multi-scale



a)

b)

Figure 4.2: Demonstration of the distance transform. a) Original image, b) its distance transform

analysis framework, based on the stability over a given pixel property, typically its lightness. The key idea of the detector we propose is to leverage the efficiency of the MSER algorithm to identify stable regions, where stability is defined as a function of the distance of a region to neighbouring ones. Hence in our framework regions that have larger distances to neighbouring ones would be more stable than regions that are close to each other.

The above algorithm is practically equivalent to a graph contraction approach, over a graph that encodes the neighbouring relationships between the connected components of the image, which in the generic case could be the fully connected graph of the connected components. A graph contraction implementation is quite inefficient. Using instead the distance transform we translate the problem from the distance domain to the image domain, where the MSER segmentation offers an efficient way to create and rank (in terms of their stability) the regions corresponding to clusters of neighbouring connected components.

4.2.1 Distance Transform

The distance transform finds the minimum distances of all image pixels to the set of foreground pixels. The result is a matrix of the same size as the image, where each element is assigned a value corresponding to the smallest distance between the corresponding image pixel and the closest foreground object.

We compute the distance transform of the document image based on the two pass algorithm that only requires linear computation time proposed in [114]. Formally, let p be a background point and q a point in the set of foreground objects Q. The distance transform $f(p)$ assigns at each background point p its distance to the nearest object point by:

$$f(p) = \min_{q \in Q} d(p, q)$$

where $d(p, q)$ is the Euclidean distance between background point p and object point q. An example of the distance transform matrix of an administrative document is shown in Figure 4.2.

Note that we implicitly assume in this discussion that the image is bi-level . However, we should point out that the distance transform concept is readily applicable to grey scale images [115].

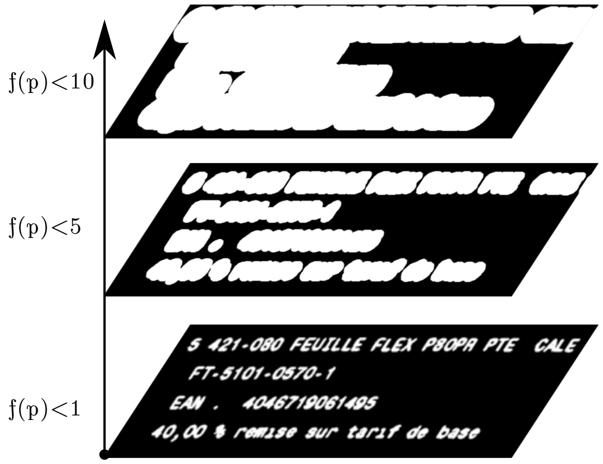

f(p)<10
f(p)<5
f(p)<1
5 421-080 FEUILLE FLEX P8OPA PTE CALE
FT-5101-0570-1
EAN . 4046719061495
40,00 % remise sur tarif de base

Figure 4.3: Example of thresholding the distance transform at different intensity levels. At the lower level individual characters can be identified, at the middle level characters have been merged into words, at the upper level words have been merged into paragraphs.

4.2.2 MSER detection

The set of maximal regions produced by the MSER algorithm is the set of all connected components produced over all possible thresholdings of the input image (essentially identical to a watershed algorithm). When calculated over the distance transform result, the maximal regions roughly correspond to semantically important structures of the document (characters, words, text lines, paragraphs), as can be appreciated in Figure 4.3.

Applying the MSER algorithm to the distance transform image produces a dendrogram of maximal regions. The leaf regions correspond to the foreground objects, while the mergers in the dendrogram depend solely on the distance between the maximal regions. An example of the typical dendrogram produced is shown in 4.4.

The MSER algorithm's δ parameter controls the minimum lifetime (number of iterations that a maximal region has to survive before merging with a neighbouring

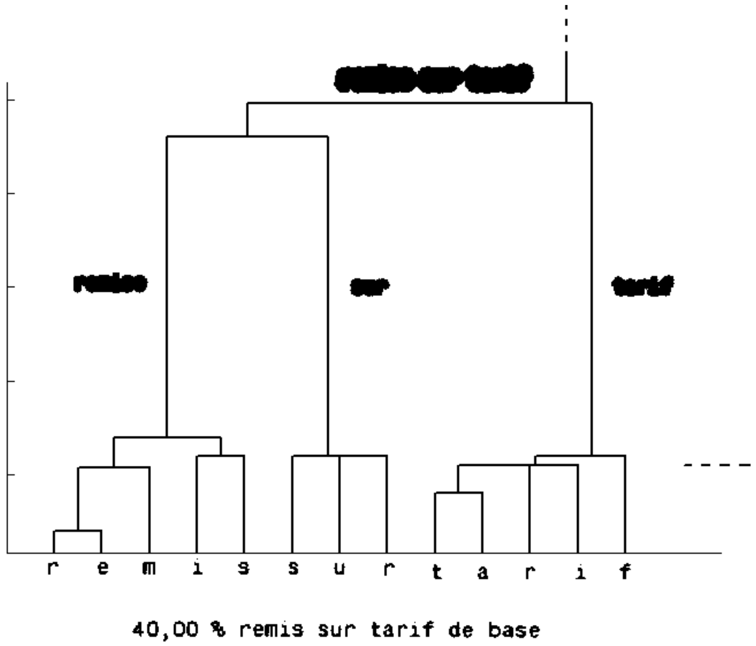


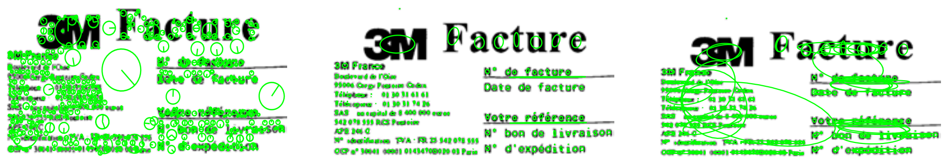
Figure 4.4: A typical dendrogram produced with the proposed method. The leaf nodes correspond to the connected components of the image, while the mergers depend solely on the distance between regions, giving rise to semantically relevant groups.

one) in order for a region to be considered stable. In the case of documents, and given the prior distance transform, δ effectively controls the minimum distance that a region has to have to a neighbouring one in order to be considered stable. As characters are the most closely positioned structures of interest, we should choose a value for δ that is less than half the minimum distance between characters, in order for them to be identified as stable. In practical terms, we can directly set $\delta = 1$, as we do not expect to have any components positioned closer to each other than characters in the document.

One potential drawback of our proposed detector is the inconsistency of the distance transformation regarding noise. However, coupling the distance transformation with a MSER analysis allows us to extract various key-regions of different sizes, only a small subset of which would be affected by such artifacts.

4.3 Experimental Results

We tested our proposed key-region detector on invoice retrieval scenario at full page level. The task is defined as searching the invoice documents that provided by the same company with query. It would indicate the system's performance on retrieving images that are structurally similar because the content (e.g. address or date) of



a) b) c)

Figure 4.5: Qualitative comparision of a) SIFT, b) MSER and c) DTMSER key-region detectors.

the invoices from the same provider might vary while the structure are consistent as they share the same template. The details and samples of the invoice dataset are described in appendix A.1. Key-regions are detected with SIFT, MSER and the proposed DTMSER methods and are subsequently described by the SIFT descriptor. For each local feature of the query, we retrieve the 100-nearest neighbours over the whole collection, each of them casting a vote at the corresponding document. For the final document retrieval, documents are sorted according to the votes received and we report the mean average precision MAP and the corresponding precision and recall plots.

We tested three different voting schemes. A *uniform* voting paradigm in which the 100 nearest neighbors give equal score to all matched documents. An *inverse rank* scoring that weights the document votes depending on their position in the nearest neighbor ranking list. Finally, a *truncated inverse distance* scoring function that equally votes for the documents that hold very small distance with the query feature and scores the rest with their inverse distance.

4.3.1 Qualitative results

We can see in Figure 4.5 a qualitative comparison of the types of key-regions identified by the three different detectors. The interest points extracted by SIFT are mainly located at letter corners. Most of the MSER produced key-regions correspond to character-level connected components. In contrast, the proposed DTMSER detector extracts multi-level features corresponding to letters, words, and paragraphs which are potentially more semantically meaningful.

4.3.2 Comparative results on a subset

We first report comparative results obtained on a subset of the database corresponding to 857 images from 50 different providers in which 10 invoices are selected as queries. To exhaustively find key-region correspondences quickly becomes infeasible when dealing with large datasets. As the SIFT and MSER detectors return an enormous amount of key-regions, we perform this first experiment on a subset of the dataset to reduce the computational cost. Furthermore, we make a use of an approximate nearest neighbour search algorithm, namely the Bucket Distance Hashing (BDH) to further reduce the computational time. Bucket Distance Hashing (BDH)

Table 4.1: MAP and time consumption for sub dataset

Detector	Num. regions	NN	Time (ms)	Voting Scheme		
				Uniform	Inverse	Truncated
SIFT	9,402,479	BF	6,148,880	0.9830	**0.9955**	0.9963
		BDH	3,640	0.9768	**0.9945**	0.9968
MSER	1,164,693	BF	135,896	0.9654	0.9667	0.9645
		BDH	679	0.9595	0.9658	0.9601
DTMSER	422,288	BF	**21,699**	**1.0000**	0.9634	**0.9990**
		BDH	**131**	**1.0000**	0.9659	**0.9984**

is a scalable approximate nearest neighbour search (ANNS) method [94]. The key idea of BDH is a combination of hash-based distance estimation and loose selection of nearest neighbour candidates, both of which are designed to find the true nearest neighbour in high probability without time consuming process. Previous experiments have shown that the BDH can reduce processing time significantly while maintaining the same accuracy as other state-of-the-art algorithms [116], a behaviour confirmed here as well.

We compare the performance of the three key-region detectors using the three different voting schemes described before. The obtained results are summarized in Table 4.1.

It can be easily appreciated that the amount of obtained key-regions is drastically reduced when using the proposed DTMSER detector instead of SIFT or MSER while the retrieval performance is not affected. There is a clear advantage in using an approximate nearest neighbour search algorithm such as BDH in the retrieval stage as there is a huge time improvement while suffering an insignificant loss in mean average precision. We show in Figure 4.6 the precision and recall plot for this experiment when using the truncated inverse distance scoring method.

4.3.3 Results on the whole dataset

To show the performance of the proposed DTMSER, we generalize our experiment over the whole invoice dataset consisting 4109 images within 249 unbalanced classes from which 383 queries are randomly picked. In this scenario the amount of key-regions returned by the SIFT and MSER detectors is very large (SIFT detects more than 40 million key-points). Therefore, we just evaluated the DTMSER detector performance combined with the BDH search algorithm.

We can see in Table 4.2 that the performance over the whole dataset is in agreement with what we obtained during the previous experiment. Regarding the different voting schemes, the truncated inverse distance strategy is the one that performs the best, although no significant differences can be observed.

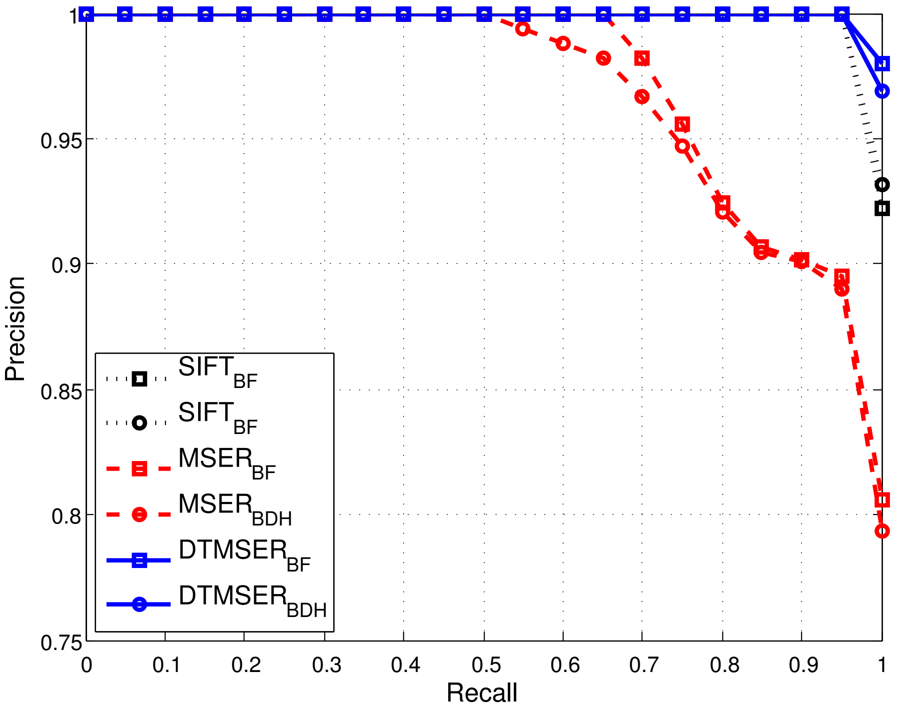


Figure 4.6: Precison-Recall curve for the sub-dataset when using the truncated inverse distance scoring method.

Table 4.2: MAP time consumption for whole dataset

Detector	Num. Key-regions	Time (ms)	Uniform	Inverse Rank	Truncated Inverse Distance
DTMSER	2,016,286	205	0.9893	0.9407	0.9909

4.4 Conclusions

In this chapter, we introduced a fast and efficient key-region detector to extract document structure as a dendrogram of key-regions where the edges represent one key-region merged to another bigger one at higher scale level. We demonstrated that the proposed DTMSER detector takes advantage of the particular structure of document images, and is able to detect semantically meaningful key-regions that roughly correspond to structural elements of the document. Compared to other state of the art detectors, DTMSER detects a much smaller number of key-regions, while achieving slightly higher performance in a retrieval scenario.

The approach followed produces a dendrogram of regions. The dendrogram produced is a rich source of structural information, as it encodes relationships between the regions. In the next two chapters, we will describe two strategies (previously proposed by us) to embed such structural information into the document retrieval process.

Chapter 5

Pair-wise BoW

In the Chapter 4, we previously discussed that DTMSER detection algorithm is capable to extract multi-scale key-regions that roughly correspond to the structural elements of the document such as characters, words and paragraphs, etc. Furthermore, we demonstrated that document structure can also be extracted simultaneously as a dendrogram of multi-scale key-regions. In this chapter, we will introduce a method to efficiently employ such explicit structure information for document image retrieval in BOW-like manner which generally achieves high performance for image retrieval problems. We will show that, benefiting from the applied explicit structural information, our method outperforms the recent state-of-the-art methods in document retrieval domain.

In the past decades, considerable effort has been made for document image classification and retrieval from different perspectives. However, generally speaking, the document images are usually represented by either their textual content [117], or their layout structure [118, 119].

Layout analysis methods explicitly describe the document structure as the spatial relations among the segmented blocks with assigned logical or physical labels. Such layout structure is usually encoded in a small number of high-level blocks (e.g. paragraphs, columns or titles) while the contents inside the blocks are ignored. The performance of layout analysis methods highly depend on the quality of image segmentation which is still a problem far from being solved. Besides, another drawback of layout analysis methods is the distance computation between groups of blocks (normally represented as graphs) since computing the similarity between graphs is widely recognized as time consuming. Such computation complexity also hinders the layout-based methods represent the document in further details (lower-level blocks such as words, or even letters). On the other hand, we focus on a document framework that is suitable for a wider range where both structure and text content features are appreciated. Consequently, layout-based methods are not considered in our research despite of its success on explicitly extracting document structure.

Document images are also widely described by their content features either globally (one feature vector per image) or locally (groups of local feature vectors per image). For example, the local key-points/key-regions are described with SIFT feature vectors in [69, 68] and as HOG feature vectors in [75, 71]. On the other hand, representing each image globally as one feature vector could achieve high efficiency for full page document image retrieval. However, neither of those representation strategies could capture the document structure which is significant important for lots of retrieval problems. Hence, based on local content description, various strategies have been proposed to compensate the drawback of lacking structural information. the most straightforward and popular option is adding spatial information like Pyramidal Bag-of Words (Pyram BoW) over Bag-of-Words (BoW).

Bag-of-Words was designed for processing text in documents and introduced by Feifei Li et al. for natural scene categorization [76] (also stated as Bag-of-Feature in computer vision domain). Afterwards, it is widely employed to represent images for various applications due to its efficiency and discriminate power. As shown in Figure 5.1, BoW representation strategy consists of 4 consecutive steps: feature detection, description, quantization and pooling. The feature detection process extracts patches (blobs or key-regions) that preserve high saliency (interest). The description process usually interprets those patches into discriminative feature vectors. In quantization step, feature vectors are assigned to nearest codewords (centroids) which are previously computed. At the end, a pooling strategy is employed to represent each image as a histogram of features whereas the basis is the assigned labels. Bag-of-Words framework interprets each image as a single histogram which preserves the discriminative power of features, thus computing the similarity between images is solved as the distance calculation of two vectors which could be implemented in extremely efficient manner.

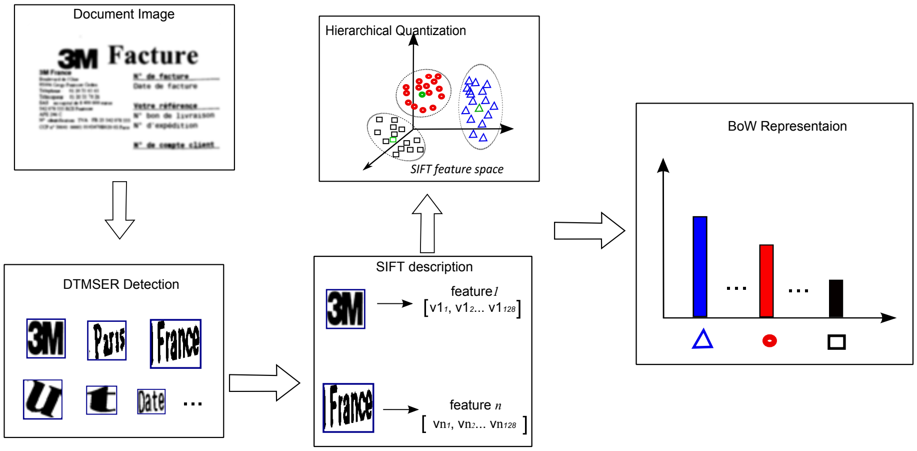


Figure 5.1: The pipeline of BoW representation for document images.

However, one significant drawback of the Bag-of-Words framework is that the spatial information of the features is not taken into account during pooling process and thus the resulted histogram vector does not convey any structural information. Several methods have been proposed to encode image structure into BoW framework while its advantage on efficiency could be preserved. Among those BoW-like methods, the most famous is spatial pyramidal BoW which is able to roughly embed the location information of each feature. As shown in Fig. 5.2, spatial pyramidal BoW iteratively divide the image into pyramidal parts and employ BoW strategy to represent each of the resulted parts. At the end, all of the BoW histogram vectors are concatenated together and employed as the representation of the whole image.

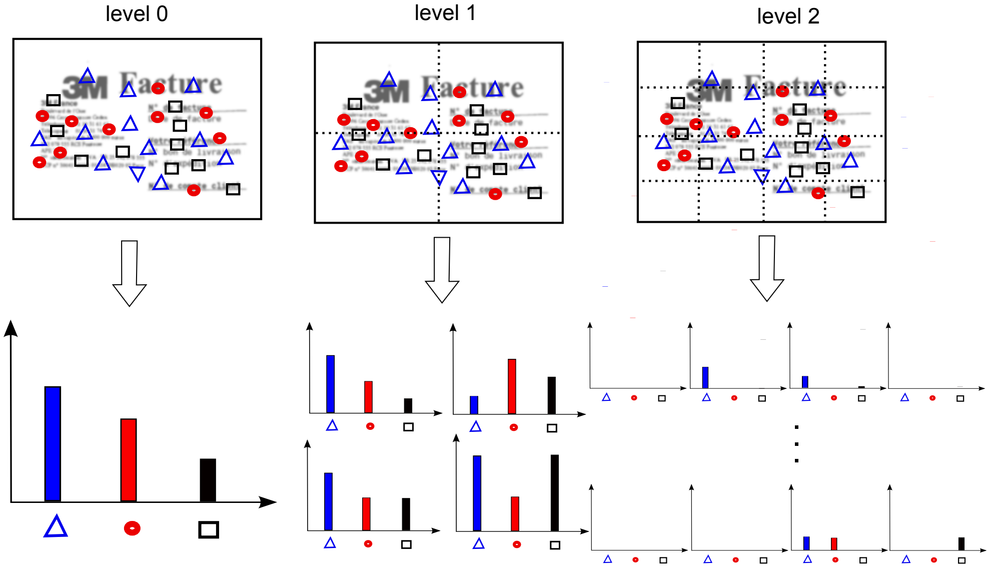


Figure 5.2: Encoding spatial structure into BoW representation

Similarly, the document image also could be iteratively divided into increasing finer sub-images and is represented as a feature vector concatenating quantized features extracted from all the resulted sub-images. A problem of the pyramidal spatial method is the dimensionality of the feature vector which increases exponentially. Besides, for document image analysis, adding such spatial information to local content feature does not explicitly encode document structure but rather the spatial distribution of local patterns.

The Distance Transform based Maximal Stable Extremal Region (DTMSER)[72] algorithm efficiently extracts the document structure as a dendrogram that roughly representing how the structural elements merge to each other (e.g. characters merge to words, words to paragraphs). The extracted dendrogram is a rich source of structural relations such as $top/down/left/right$, *neighboring* within specific distance and *inclusion*. Nevertheless, how to query such explicit structural relations in efficient manner is still under challenge. In this chapter, we propose an approach to dace this challenge.

In this chapter, we will present an efficient method, namely pair-wise BoW, that

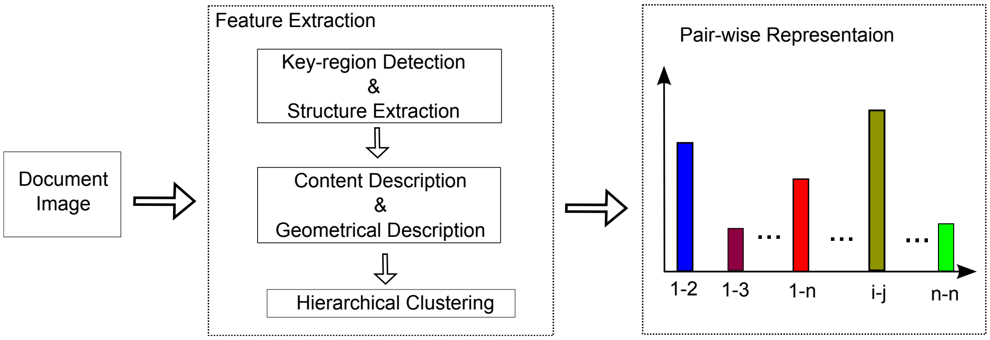


Figure 5.3: The pipeline of document retrieval based on the proposed pair-wise representation.

incorporates the BoW method with the *inclusion* structural information which commonly exists between structural elements of document images. The main advantage of our method is that the explicit document structure is efficiently embedded into a BoW framework through pair-wise key-region representation. As illustrated in Figure 5.3, we extract the document structure as a dendrogram of key-regions containing rich source of structural relations. Afterwards, each key-region is described by two types of features: geometrical feature and content feature. To generate the codebook, hierarchical k-means algorithm is then employed to quantize the geometrical and local content features in two consecutive stages. At the end, the document dendrogram is ”decomposed” into list of key-region pairs (edges in the dendrogram) which are employed during pooling process to generate the BoW histogram representation. Briefly speaking, pair-wise BoW represent each document into a histogram where the basis is key-region pairs while the basis of the original BoW representation is separated key-regions. The pair-wise BoW representation expresses the statistical characteristic of key-regions where *inclusion* structural relation is encoded.

However, the pair-wise BoW representation strategy leads to higher dimensionality that equals to the square of codebook size. Nevertheless, we observe that the histogram vector is very sparse. Consequently, to solve such problems, inverted file indexing strategy is employed in order to efficiently compute the distance between the sparse histograms (will be explained in Section 5.2).

5.1 Feature Extraction

Formally, a document D is represented by a tree structure $T(R, E, l)$ where $R = \{r\}$ is a set of tree nodes corresponding to key-regions extracted by the DTMSER algorithm (see section 5.1.1), $E = \{e\}$, whereas $e = (r\_i, r\_j)$ and $r\_i, r\_j \in R$, is a set of directed edges in the tree representing *inclusion* relations, and l is a labeling function assigning shape attributes regions $l : R \rightarrow \mathbb{R}^n$. For each node r, $l(r)$ is employed to map its feature vector $f(r)$ to an integer (numeric label).

5.1.1 Distance based MSER (DTMSER)

The DTMSER algorithm, proposed in previous work [72] takes advantage of the fact that the structure of a document is tightly linked to the distance among its elements: characters are located closer to each other than words are, which are in turn placed closer to each other than paragraphs are. As shown in Fig. 5.4, DTMSER algorithm basically comprises two steps: distance transform and MSER analysis.

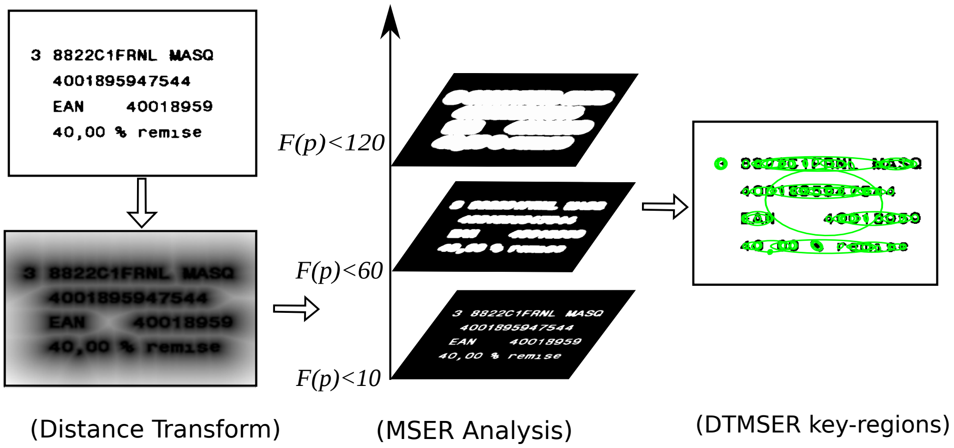


Figure 5.4: Progress of Distance Transform MSER algorithm.(a) Distance transform; (b) Thresholding process for MSER analysis; (c) the resulted key-regions represented as ellipsoids and a dendrogram linking the key-regions together.

For each pixel p in the image, distance transform algorithm basically set its value as the minimum distances to the set of foreground text pixels. Such "preprocessing" step transforms the image from x-y space into distance space where the more explicit borders could be observed among different scales of document structural elements, because characters are usually located closer to each other than words are, which are in turn placed closer to each other than paragraphs are.

In order to produce a distance-aware version of MSER, we take the distance transformed image as input and compute the classical MSER detection over it in order to identifies the stable regions if their size change is small during the thresholding process as shown Fig. 5.4(b). We set the thresholding strand (namely delta in the algorithm) to 1 and thus build a redundancy tree $T(R, E)$ that contains all the regions r generated by thresholding process. Let us denote r_{i-1} and r_{i+1} as the immediate child (if exist) and parent of the ith region r_i, $Area_i$ and $Area_{i+1}$ as the sizes of the regions r_i and r_{i+1} respectively. The initiative tree is truncated into a maximally stable extremal region tree by deleting the (ith) node if $(Area_{i+1} - Area_i)/Area_i > 1$ (unstable) and linking r_{i-1} and r_{i+1} together: $\{e(r_{i-1}, r_i), e(r_i, r_{i+1})\}$ to $\{e(r_{i-1}, r_{i+1})\}$. Afterwards, the stable key-region tree is further simplified by filtering out the duplicated MSER regions. For example, the key-region r_i is recognized as duplicated and thus deleted if $(Area_{i+1} - Area_i)/Area_{i+1} < 0.5$ while r_{i-1} and r_{i+1} is directly linked together afterwards. In our experiment, such strategy gives us a dendrogram on multi-level key-regions that roughly correspond to semantical elements of the document (letters, words, paragraphs) since due to typesettings the distances between these elements correspond to local maxima in the distance transform.

DTMSER detector leverages the efficiency of the MSER algorithm to extract stable regions. As shown in Figure 5.5(a), DTMSER algorithm is capable to efficient extract document structure as a dendrogram of multi-scale semantical key-regions that roughly correspond to letters, words, paragraphs.

5.1.2 Descriptors

For each DTMSER key-region r, affine normalization is employed to transform the corresponding region to square size facilitating content feature description afterward. We will test three different algorithms, SIFT, HOG and Run Length (RL), to compute the content feature of each key-region $f_c(r)$.

- **SIFT descriptor** divides each normalized region into a 4×4 grids. Eight bins are used to quantize gradient vectors, then a histogram of 8 bins describes each grid. Hence, SIFT descriptor represents each region as a feature vector with $4 \times 4 \times 8 = 128$ dimensions. However, Gaussian weighting process is ignored in our case because the text close to the boarder is considered as important as the central part and a slight improvement is observed during the experiments.

- **HOG descriptor** computes features by dividing each normalized region in 4 by 4 cells and then 31 features are extracted from each cell [56]. At the end, $4 \times 4 \times 31 = 496$ dimensional feature vector is returned for each key-region.

- **RL descriptor** also computes the histogram of the content features but in terms of run length which is defined as the number of pixels with the same value in a sequence. As discussed in [84], we quantize run lengths in a logarithmic manner into 9 bins as follows: [1], [2], [3-4], [5-8], [9-16], [17-32], [33-64], [65-128], [129-Inf]. For binary images in our case, run length yields $2 \times 9 = 18$ bins for both black and white sequences. Besides, we compute runlength feature in horizontal, vertical, diagonal and anti-diagonal directions resulting in $4 \times 18 = 72$ dimensions in the final feature description.

One drawback of affine normalization applied to content description is that it discards the geometrical information (e.g. aspect ratio of the bounding box) of the original key-regions. Hence, the SIFT descriptor that takes the squared regions as input, is not capable to discriminate the geometrical characteristics from one key-region to the other. To compensate such loss, we proposed to further describe key-region r by its geometrical feature $f_g(r)$ which is represented in two dimensions: the aspect ratio of the bounding box and the area ratio between key-region and the corresponding bounding box. Another reason to apply $f_g(r)$ is that it is more flexible on content variation and thus allows our algorithm to mainly use structural information and solve the structure spotting task.

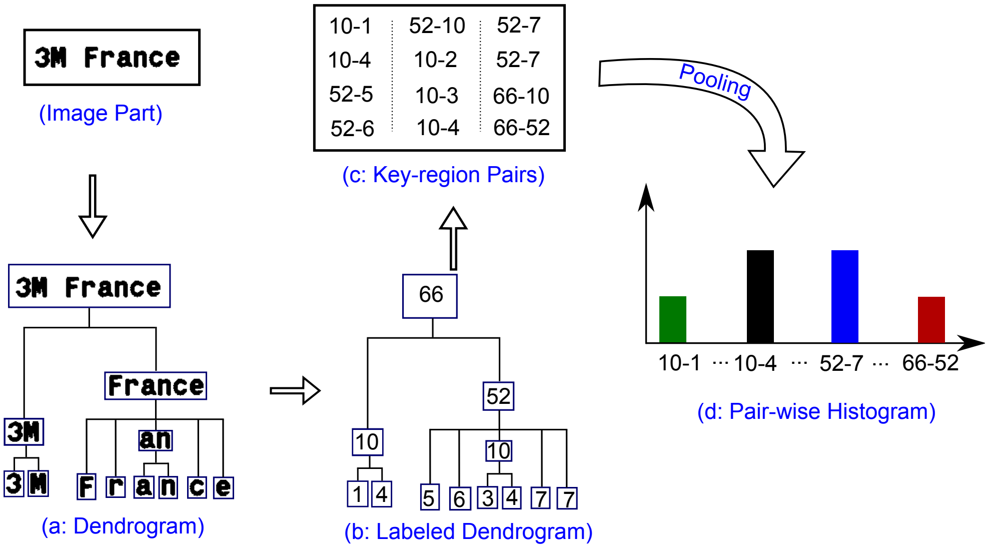


Figure 5.5: Pair-wise BoW pipeline with details.

5.1.3 Codebook

To create the codebook of the key-regions, we employed a hierarchical clustering strategy applied in two consecutive steps: first geometrical feature quantization $l_g(r) : f_g(r) \to int_g$ and then content feature (SIFT) quantization $l_c(r) : f_c(r) \to int_c$. K-means clustering algorithm is employed in both steps. The final label is assigned as $l(r) = [l_g(r), l_c(r)] = int_g \times n\_des + int_c$ where n_des denote the number of centroids for content feature. Thus, the codebook size equals to the product of the number of centroids for $l_g(r)$ and for $l_c(r)$. For example, assume the number of centroids for geometrical and content feature are configured as $n\_geom = 10$ and $n\_des = 100$, the codebook size would be: 10*100=1000. In section 6.2.2, we study how the performance changes when the importance of the two types of features varies while the respective cluster numbers are configured as $n_geom \in \{5, 10, 15, 20, 25\}$and $n\_des \in \{50, 100, 150, 200, 250\}$ resulting in 25 parameter configurations.

In summary, as shown in Figure 5.5(b), after key-region detection, feature description and codewords assignment, a document image is represented as $T(R, E, l)$. Afterwards, by applying a consecutive two-step quantization process, a numeric label is assigned for each key-region $r \in R$ according to the corresponding features: content feature $f_c(r)$ and geometrical feature $f_g(r)$. More importantly, the edges of the tree $e = (r_i, r_j) \in E$ carry lots of structural relationships such as *intersection*, *top/left* and *inclusion*. Considering the simplicity of the image representation, we only consider the edges in parent-child manner that convey *inclusion* relations.

5.1.4 Pair-wise Pooling

We employed DTMSER to extract the document structure as a hierarchical key-region tree $T(R, E)$ where each $r \in R$ is quantized to its nearest codeword in the feature space. The main promising advantage of such key-region tree is that its edge $e(r_{i-1}, r_i) \in E$ contains a rich source of *inclusion* structural relations between the parent node and child node. On the other hand, BoW framework efficiently represent each image as a histogram of separate key-regions but structural relations among the key-regions is ignored. Even though spatial pyramidal BoW representation could partially carry the structural information by encoding the rough location of each key-region, such spatial structure only encode the local patterns rather than document structure and would fail when image rotation occurs. Hence, it is necessary to encode other structural relations (e.g. *inclusion*) which are explicit and rotation invariant. Consequently, we proposed to "decompose" the labeled dendrogram into lists of key-region pairs (edges) and represent each image as a BoW -like histogram while the pooling elements are key-region pairs. The main advantage of our method is that we manage to embed *inclusion* structural relation into the pooling elements while the standard BoW use orderless separate key-regions without any structural information as pooling elements.

The problem of the proposed method is that the size of the codebook is squared. For example, assume the numbers of clustering centroids for geometrical and content feature are set to be 20 and 200 respectively, the codebook size of standard BoW would be $s\_codebook = 20 \times 200 = 4000$. However, in the case of the proposed pair-wise BoW representation, the codebook size would be $s\_codebook = 4000 \times 4000 = 16 Million$.

5.2 Inverted File Indexing

The higher dimensionality of the pair-wise BoW representation would lead to the increased computation complexity for calculating the similarity between images. To address this problem, we apply Inverted File Indexing (IVF)[120] which is independent to codebook size for calculating the similarity between two images.

The proposed method is applied for retrieving images based on structural information, allowing for slight variation on key-region locations. As an example consider in an administrative application, for invoice images from the same provider, the logo location may change from one document to another. Hence the homography calculation process that is usually employed to check the spatial consistency of the matched local patterns is ignored in our case. This strategy could significantly reduce the required key-region storage space and the time consumption of query process. As showed in Figure 5.6, the words are stored with the image id and its occurrence time represented here as *count*. During query time, the distance calculation process is only performed for the database images that have at least one matched key-region pairs while other images that do not share any key-region pair are directly ignored. Since the codebook size of the proposed pair-wise method is the square of the standard BoW method, the corresponding histogram vector would be very sparse. Hence, when computing the distance between query and target images, only the non-zero dimension in their representation vector is actually computed. As argued in [121], we employ Cosine

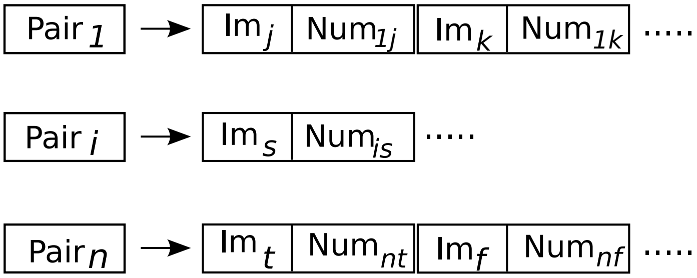


Figure 5.6: The Inverted File Indexing structure of the key-region pairs.

distance to calculate the dissimilarity of two images while L2 normalization process is performed in advance. To give more importance to the rare key-region pairs which are more discriminative, the *tf-idf* (Term Frequency - Inverse Document Frequency) [122] weighting scheme is applied.

5.3 Experiments

We apply the proposed method to an invoice retrieval scenario at full page level (see Appendix A.1). Overall, 4.7 million multi-level stable key-regions are extracted by the DTMSER algorithm corresponding to approximately 1000 key-regions per image on average.

To obtain the ground truth, we assume that two images would be structurally similar if they come from the same provider and they would be different if they come from different providers. Mean Average Precision (MAP) is employed here to evaluate the performance of the proposed pair-wise BoW. Since our method is a variant of BoW framework, we consider the BoW and spatial pyramidal BoW that widely applied for document retrieval as the baseline.

The experiment is discussed in two parts: 1)parameter validation on number of clustering centroids and different type of content feature descriptor (SIFT, HOG and RL); 2)then performance comparison of the proposed method with BoW and Spatial BoW is discussed afterwards.

5.3.1 Parameters Validation

The parameters such as the numbers of clustering centroids for geometrical feature and content feature determine the size of codebook. The codebook size controls the discriminative power of the assinged label and thus would significantly affect the retrieval performance. Consequently, to figure out the optimal parameter configuration, a validation process is performed on the number of centroids of the two considered features and content feature description strategies (SIFT, HOG, RL).

To fairly compare the performance of the proposed method with BoW and spatial BoW in Section 5.3.2, the validation process is also performed for the two considered baseline methods. The corresponding results of BoW, spatial BoW and the proposed

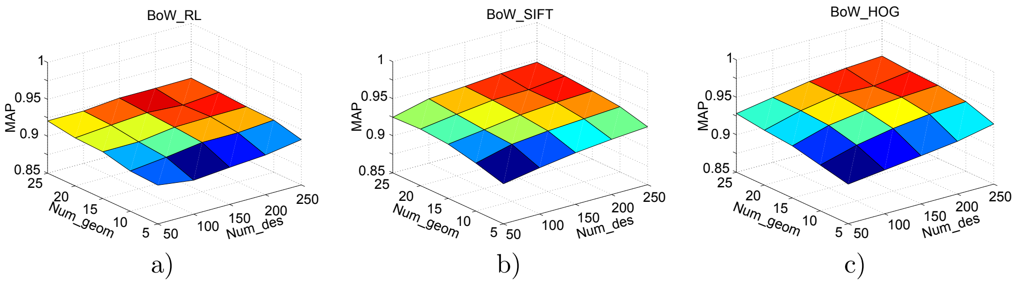


Figure 5.7: Clustering parameter Validation of a) Run Length, b) SIFT and c) HOG descriptor based on BoW.

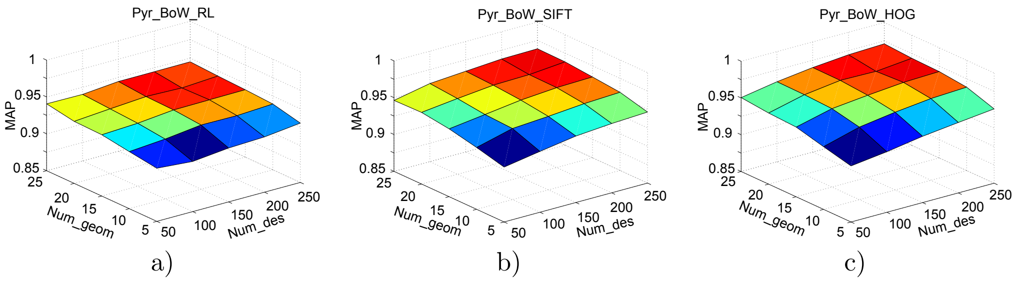


Figure 5.8: Clustering parameter Validation of a) Run Length, b) SIFT and c) HOG descriptor based on Pyramidal BoW .

method are illustrated in Figure 5.7, 5.8, 5.9 respectively.

We represent the combination of number of geometrical and content feature clustering centoirds as Num_geom and Num_des for short respectively and the two parameters is configured as $Num_geom \in \{5, 10, 15, 20, 25\}$and $Num\_des \in \{50, 100, 150, 200, 250\}$ resulting in 25 parameter combinations. As demonstrated in Figure 5.7, 5.8, 5.9, despite of the descriptor types and the retrieval methods, within the considered range, increasing the number of centroids of either geometrical or content feature will result in a performance improvement. That is because increasing the number of clustering centroids actually leads to the enhanced discriminative power of corresponding features. However, for structural retrieval, this does not indicate that the bigger the codebook size is the better performance would be since the feature may become to be too discriminative. Even though, we could observe that, when the $Num\_geom > 15$ and $Num\_des > 150$, increasing the number of centroid (either Num_geom or Num_des) does not lead to obvious improvement on retrieval performance indicating that the ceiling point is most probably reached. However, taking the slight performance improvement into account, we choose $Num\_geom = 25$ and $Num\_des = 200$ as the optimal configuration for the number of clustering centroids. Besides, since inverted file indexing is applied here, increasing the number of centroids does not lead to higher (actually slight less) computation complexity.

Among the considered content descriptors, for most cases, RL performs worst and HOG performs best. This makes sense because RL simply encodes the information

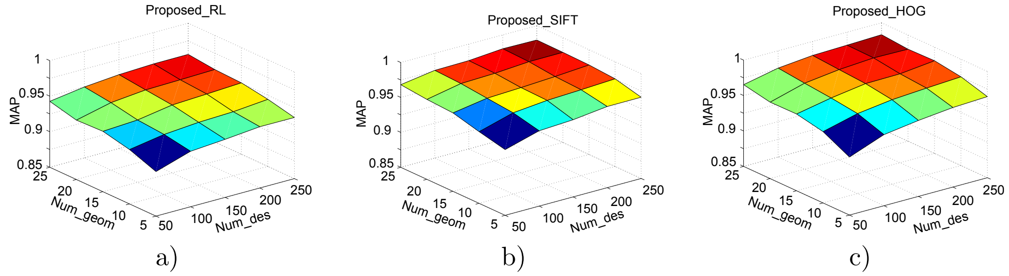


Figure 5.9: Clustering parameter Validation of a) Run Length, b) SIFT and c) HOG descriptor based on the proposed method.

about number of object pixels which is less discriminative for representing the local content than the gradients information that employed by both SIFT and HOG. For the same type of information (SIFT and HOG), increasing the dimensionality would probably lead to the enhancement of discriminative power. Consequently, the RL descriptor with less discriminative power performs worse than the SIFT and the HOG descriptors. Taking the advantage of higher dimensionality, the HOG descriptor achieves the slightly better performance than the SIFT descriptor. Generally speaking, SIFT obtains more than 2% better performance than RL descriptor and around 1 % worse performance than HOG descriptor. Considering their dimension and the resulted computation complexity for assigning labels, SIFT is recognized as the best descriptor here even it performs 1 percent less than HOG. Because at 4 times calculating time for label assigning process resulting in 1 percent better performance is not ”economic” especially in the case of large scale retrieval.

In conclusion, $Num\_geom = 25$, $Num\_des = 200$ and SIFT descriptor is considered as the optimal configuration for BoW, spatial BoW and the proposed method.

5.3.2 Proposed VS BoW

In this section, we compare the retrieval performance of our method with BoW and spatial BoW based on the parameters validated in the previous section. Both MAP and *precision-recall* curve is employed to demonstrate the performance difference.

Table 5.1: MAP performance of descriptors and frameworks($n\_geom = 25$ and $n\_des = 200$)

	BoW	BoW_Pyram	Proposed
RL	0.9254	0.9448	**0.9559**
SIFT	0.9444	0.9630	**0.9802**
HOG	0.9493	0.9693	**0.9816**

Concerning RL, SIFT and HOG descriptor, table 6.2 shows the performance of the considered retrieval methods. As argued in section 5.3.1, despite of the retrieval methods, SIFT generally obtains 2% better performance than RL and less than 1%

worse performance compared to HOG. Among all the considered retrieval methods, benefiting from the pyramidal spatial information, spatial BoW achieved around 2% improvement over the standard BoW method. Benefiting from the explicit structure of document images, the proposed method gains further 2% better performance than spatial BoW which represents the document's structure implicitly as spatial distribution of local patterns. The *precison-recall* curve of three compared retrieval methods is plotted in Figure 6.9 based on SIFT descriptor and optimal number of clustering centroids.

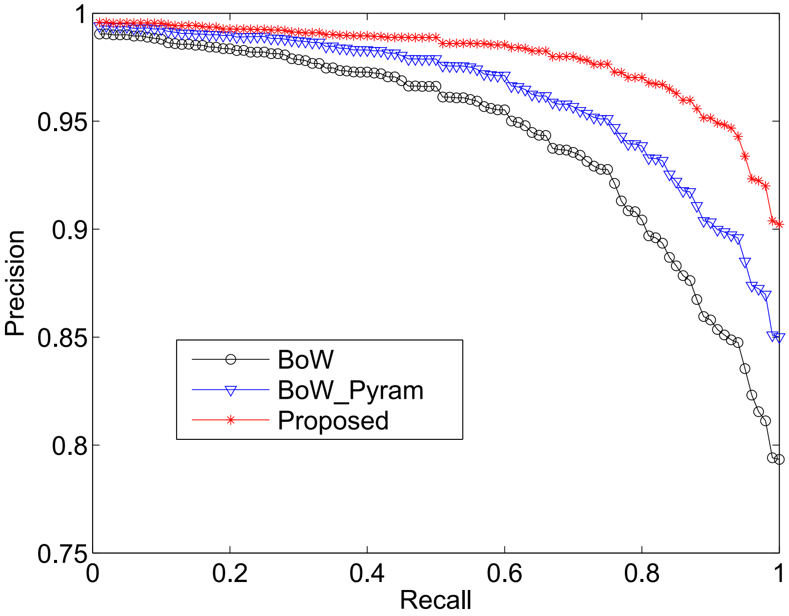


Figure 5.10: Precision-Recall curve of SIFT descriptor based on BoW, Pyramidal BoW and the proposed method.

5.4 Conclusion

In this chapter, we have presented an inverted file indexing based method for structural document image retrieval. The document image is represented as a list of paired multi-level stable key-regions which generally corresponding to character-word or words-paragraph pairs with *inclusion* structural information explicitly incorporated. Instead of pooling separate key-regions to generate the histogram representation in the case of the BoW method, we employ the key-region pairs that carry *inclusion* structural information as pooling elements. The inverted file indexing strategy is employed to solve the computation complexity problem caused by quadratic codebook size.

Under the full page invoice image retrieval scenario, we compared the performance of the proposed method with BoW and spatial BoW method while a validation process on content feature descriptor and number of clustering centroids of both geometrical and content feature is performed. We demonstrated that, within the considered BoW-like methods, spatial pyramidal BoW achieves better results by taking the implicit spatial information into account. Furthermore, benefiting from the explicit *inclusion* structural information that encoded in the key-region pairs, our proposed pair-wise methods gains further improvement comparing with spatial pyramidal BoW.

However, the proposed pair-wise BoW framework is not able to handle the focused retrieval scenarios where the partial areas of images (instead of the whole images) are expected from the collection. Such limitation would narrow down the employment of the demonstrated explicit structure. Hence, in next chapter, we will introduce a generic framework that allows to apply the *inclusion* structural relation in the focused retrieval scenario.

Chapter 6

Spatial Database

In the previous chapter, we introduced an efficient framework, named as pari-wise BoW, which represents document images as a histogram while the pooling elements are key-region pairs instead of separate key-regions. We demonstrated that, benefiting from the explicit *inclusion* structural relation between key-regions pairs, the pair-wise BoW methods achieves further improvement against spatial pyramidal BoW that widely employed for document image retrieval.

Even though such global representation was illustrated to be very efficient and effective for retrieving document images at full page level (see Figure 6.1(b)), it is very challenging to apply representation for retrieving the correspondences when only a small areas/portions of the images are expected. For example, both object detection for PASCAL challenges [123] in computer domain and logo retrieval (see Figure 6.1(a)) in document community seek to further locate the counterparts inside given images.

Besides, from the viewpoint of similarity, documents comprise a particular type of images where structure is quite explicit, and at the same time decoupled from content. For example address blocks on letters, share the same structure across different letters, although the exact content might change. Other examples might include newspaper articles, figures with their respective captions, or items on an invoice as shown in Figure 6.1(c).

Discovering similar areas in images is of great practical interest, and it makes intuitive sense to include structural information in such a search. In the past decades, document retrieval problem have been treated at either full page or part-based (parts of image) level where the structure or visual similarity is measured. However, for many applications it would make sense to try to decouple visual features and structural information to allow conducting searches based on structural similarity solely, or in the generic case to put a desired weight on content versus structure. The expected benefit would be an extension of current methodologies to the discovery of areas that share the same structure while the local content might vary, going over and above discovering exact matches only.

To offer a practical example from the document image analysis domain, this would translate to the ability of discovering e.g. address blocks in letters given a single address block as a query on the basis of their common structure, ignoring possible

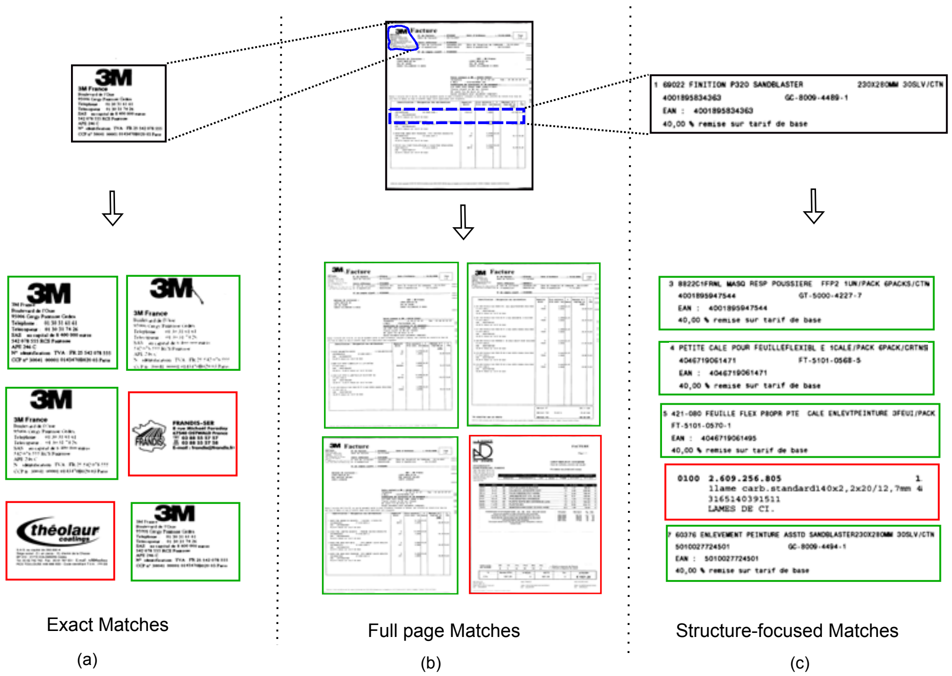


Figure 6.1: Document Retrieval in three different scenarios: (a) Exact matches aiming to retrieve all the image parts that preserve the content and structural similarity; (b) Full page retrieval searching for both content and structurally similar images; (c)Structure-focused matches expected to return the image parts where the structural similarity is preserved but the content could be differ.

content variations between address blocks. This situation, made easy to study in the realm of document images, is nevertheless quite generic, and solutions to this problem could potentially have a use in every application where a part based model is used. It should be stressed that in the context of this work, we use ”part-based retrieval” to refer to the retrieval of areas (parts of images) instead of full images, and not to the use of any part-based model.

This is an open research problem, that departs from exact object matching and verges upon structural pattern recognition approaches and in particular sub-graph matching techniques (assuming a consistent graph representation is easy to define). The main downside with employing a fully fledged sub-graph matching approach is its high computational cost that forbids its use in real-time applications.

The work presented here addresses the problem described above from a fresh perspective, proposing a generic solution for structure based retrieval. As shown in Figure 5.3, we represent query image as a group of key-region pairs that correspond to the edges in the dendrogram tree. Correspondences are retrieved for each key-region pair if the matches on both visual features and structural relations are observed. At the end, the areas/parts of the target images that holds grant number of correspondences are located as the matches of the query image while RANSAC is employed to check the spatial consistency.

The methodology is generic, in the sense that the relative contribution of visual versus structure can be flexibly adjusted (hence allowing the same method to tackle the whole spectrum of problems from exact visual based matching to purely structure based retrieval), while it can be used within a local, spotting framework as well as within a global, full image retrieval one. The main focus of this part of work is on the more complex problem of retrieving local structure similarities in collections of images given a local structural query, although we demonstrate how the exact same framework can be used for full-page retrieval scenario yielding state of the art results.

The main innovative points of our work are as follows. We propose the use of spatial databases for efficient indexing and retrieval of local structural inclusion patterns. We propose a generic technique for structure based retrieval making use of inclusion relationships and spatial indexing. We evaluate the proposed method on a dataset of administrative documents and demonstrate that flexible retrieval of local regions based on structural similarity is possible without penalising performance in exact (visual-based) matching scenarios. We finally demonstrate that the framework can be used as is for full-page retrieval, yielding state of the art results.

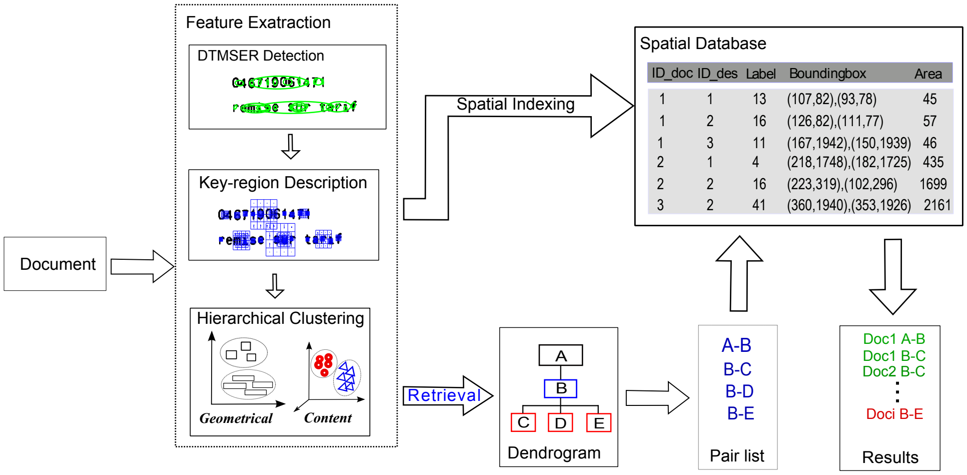
6.1 Pair-wise spatial retrieval

6.1.1 Document representation

As discussed in Chapter 5, we employed DTMSER detection algorithm to interpret each document image as a dendrogram $T(R, E)$ where E represents set of edges $e = (r_i, r_j) \in E$ and R indicates group of key-regions (nodes) $r \in R$ that roughly corresponds to the structural elements of the documents (letters, words, paragraphs, etc.). Afterwards, each key-region r is described by its visual features which consists of geometrical features $f_g(r)$ and content features $f_c(r)$. A consecutive k-means clustering is performed in order to create codebook for geometrical and content feature separately. Afterwards, hard assignment is applied to assign label for each key-region as $l(r) = l_g(r) \times num_c + l_c(r)$ where num_c is codebook size of content feature and the $l_g(r)$ and the $l_c(r)$ represent the IDs of nearest codewords of its geometrical and content feature respectively. As the result of the assignment process, the visual features of each key-region are quantized and thus the document is further described as $T(R, E, l)$ where l indicates the label information of each key-region. At the end, each document is represented as a group of key-regions pairs (edges e)which is employed as the pooling elements to generate the pair-wise BoW histogram.

However, pooling the key-region pairs into a histogram and globally represent each document as a vector is not suitable for focused retrieval that aims at to further locate portion of image as a correspondence. Hence, we will alternatively introduce a new strategy which directly retrieve corresponding key-region pairs instead of pooling them into a histogram.

As shown in Figure 5.3, similarly with pair-wise BoW framework, we extract a key-region tree (dendrogram) $T(R, E, l)$ for the query and represent the image as a group of key-region pairs $e = (r_i, r_j)$. However, during query time, for each key-



ID_doc	ID_des	Label	Boundingbox	Area
1	1	13	(107,82),(93,78)	45
1	2	16	(126,82),(111,77)	57
1	3	11	(167,1942),(150,1939)	46
2	1	4	(218,1748),(182,1725)	435
2	2	16	(223,319),(102,296)	1699
3	2	41	(360,1940),(353,1926)	2161

Figure 6.2: Pipeline of the proposed structural matching.

region pair, we directly retrieve all of its counterparts $e\prime = (r_i\prime, r_j\prime)$ where the labels of corresponding key-region match and meanwhile the same structural relation is observed in $e = (r_i, r_j)$ and in $e\prime = (r_i\prime, r_j\prime)$. However, comparing the relative spatial relations (e.g. $left/top$ and $inclusion$) between key-regions would be very expensive. Hence, it is essential to incorporate with proper indexing strategies to boost such pair-wise structural querying process.

In document analysis domain, plenty of indexing methods have been proposed such as Approximate Nearest Neighbors (ANN) [90], Locality Sensitive Hashing (LSH) [91], inverted file indexing [92], Product Quantization (PQ) [93], Bucket Distance Hashing (BDH) [94], k-d tree [95, 85] have been proposed or applied to document image retrieval domain. But neither of them could handle the relative spatial relations between key-regions.

On the other hand, spatial databases [96] are designed for dealing with the spatial relations (*inclusion*, *intersection*, *overlapping* etc.) among geometrical objects such as points, lines, polygons, etc. In the past decades, lots of spatial indexing techniques have been proposed for spatial databases to roughly 'memorize' the spatial relations among all stored objects. Hence, spatial databases allow to cast queries in terms of geometrical relationships among the stored objects in efficient fashion. For example such databases support queries such as "*retrieve all the objects having a border close to point **A** that overlap with circle **B** and intersect with the polygon **C***". They have been widely used in various Geographical Information System (GIS) applications such as maps, national census, car navigation, global climate change research, etc. however, to our best knowledge, they have not been exploited in the document analysis community up to now.

In order to boost the time consumption of spatial querying, various techniques such as R-tree, R^+-Tree, R^*-tree are designed for indexing the stored objects based on their minimum bounding rectangle (MBR). Taking R-tree indexing strategy as an

The transformation approach [Hi85, SeK88], here shown with the *corner representation*, generally leads to rather skewed distributions of points. For example, all points fall into the area above the diagonal $x = y$. If all intervals are small, all corresponding points lie very close to this diagonal. It is also possible to use a *center representation* (using center and length of an interval) but then the query regions become cone-shaped which does not fit so well with rectangular partitions of the point set. The LSD-tree point data structure was designed particularly with the goal to be able to adapt to such skewed distributions [HeSW89]. A recent discussion of the transformation approach and a comparison to methods storing rectangles directly can be found in [PaST93].

Overlapping regions. The prime example of a structure using overlapping bucket regions is the R-tree [Gu84], illustrated in Figure 11.

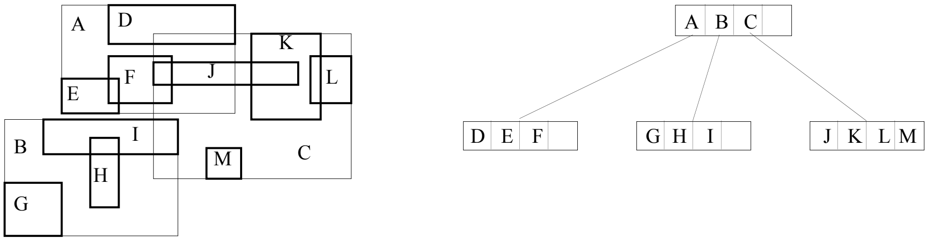


Figure 11: A set of rectangles represented by an R-tree

It is a multiway tree, like the B-tree, and stores in each node a set of rectangles. For the leaves, these are the rectangles of the set R to be represented. For an internal node, each rectangle is associated with a pointer to a son p and represents the bucket region of p which is the bounding box of all rectangles represented within p. For example, in Figure 11 the root node contains a rectangle A which is the bounding box of the rectangles D, E, and F stored in the son associated with A. Rectangles may overlap; hence, a rectangle can intersect several bucket regions but will be represented only in one of them. An advantage is that a spatial object can be kept in just one bucket. A problem is that search needs now to branch and follow several paths whenever one is interested in a region lying in the overlap of two son regions. To keep search efficient, it is crucial to minimize the overlap of node regions. This is determined by the split strategy on overflow. Several strategies based on different heuristics have been studied in [Gu84, Gr89, Beck90]; the one proposed in [Beck90], called *R*-tree*, appeared to perform best in experiments.

Clipping. A variant of the R-tree, called R^+*-tree*, was proposed by [SeRF87, FaSR87] and used in the PSQL database system [RoFS88]. It avoids overlapping regions associated with buckets on internal nodes of the same level completely by clipping data rectangles, if necessary.

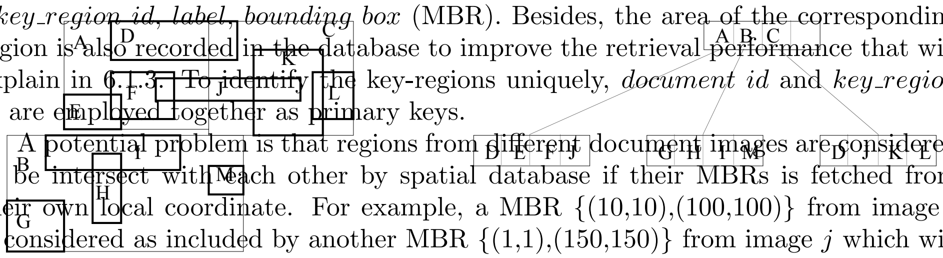


Figure 12: A set of rectangles represented by an R^+-tree

example, as showed in Fig. 6.3, the MBRs (**D, E, F**) included in another MBR (**A**) are placed in the child branch of their outbound MBR. The most straightforward advantage of R-tree indexing is the computation reduction when querying the regions that lie within given region because only the regions whose MBRs are included by the MBR of the given region are checked while other regions are directly ignored. The further details of R-tree indexing technique could be found in [124].

Figure 6.3: R-tree indexing for spatial database.

6.1.2 Spatial Indexing

As stated before, structural relationship, such as *top/left*, *inclusion*, commonly exists on each edge $e = (r_i, r_j) \in E$. More abstractly, the extracted document structure is a dendrogram tree composed by labeled objects linked those structural relationships. On the other hand, spatial databases are capable to efficiently deal with lots of relative location relationships among objects. Hence, we propose to index the structural/spatial relations among the document key-regions $r \in R$ through spatial database. We observed that such indexing strategy would significantly improve the efficiency when comparing the spatial relations (especially for *inclusion* relation that employed in this work).

PostgreSQL database system[1] is adopted to store regions into spatial database. Its schema showed in table 6.1. Each key-region is stored as a record consisting of *document id* , *key_region id*, *label*, *bounding box* (MBR). Besides, the area of the corresponding region is also recorded in the database to improve the retrieval performance that will explain in 6.1.3. To identify the key-regions uniquely, *document id* and *key_region id* are employed together as primary keys.

A potential problem is that regions from different document images are considered to be intersect with each other by spatial database if their MBRs is fetched from their own local coordinate. For example, a MBR {(10,10),(100,100)} from image i is considered as included by another MBR {(1,1),(150,150)} from image j which will not happen in reality if $i \neq j$. To avoid this, we define coordinate globally by aligning images one besides another along X-axis as showed in figure 6.4.

Spatial indexing, called GiST index which is an hybrid implementation of B-tree, R-tree and many other spatial indexing schemes, is built on all the stored regions

[1] We employed PostgreSQL software which is a powerful, open source spatial database system

Table 6.1: Data Structure Stored in Spatial Database.

Document id	Key-region id	Label	Bounding box	Area
1	1	680	(107,82),(93,78)	56
1	2	898	(126,82),(111,77)	75
1	3	616	(167,1942),(150,1939)	51
2	1	59	(1718,1748),(1682,1725)	828
2	2	893	(1723,319),(1602,296)	2783
3	1	858	(3267,82),(3251,78)	64
3	2	460	(3281,2202),(3214,2128)	4985

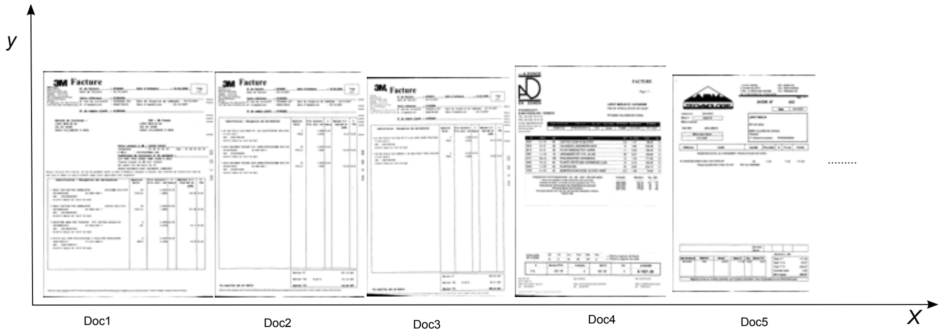


Figure 6.4: Image coordinate definition.

based on their corresponding MBRs' location. All the extracted key-regions $R = \{r\}$ are stored separately and spatially linked together by the tree-structured index.

6.1.3 Structural Retrieval

The DTMSER algorithm efficiently computes the document structure represented as $T(R, E, l)$. Taking advantage of the spatial indexes, the system could efficiently query any type of spatial structures between key-regions (r_i, r_j) such as *top of* or *left of* or *within given distance* etc. To reduce the complexity of similarity computation between the querying $T(R, E, l)$ and the stored document, we propose to represent the document structure as a list of edges $e = (r_i, r_j)$ with *inclusion* relationships between the two associated key-regions. Then the edges are employed to retrieve their matches from spatial database like *find all the possible edges* $e\prime = (r_{i\prime}, r_{j\prime})$ *where the corresponding labels* $l(r_i) = l(r_{i\prime})$, $l(r_j) = l(r_{j\prime})$ *and meanwhile* $r_{i\prime}$ *is included by* $r_{j\prime}$. Spatial indexing employed here can significantly boost such edge retrieval process. RANSAC algorithm is then employed to check the global consistency among the returned edges and a ranked retrieval list is returned afterwards based on the number of inliers.

The main advantage of the proposed pair-wise key-region querying against isolated orderless key-region querying is that it is capable to efficiently query many structural information such as $top/left/\ of$, *intersection* and the *inclusion* relationships that employed in our research. However, when retrieving the edges $e' = (r_i', r_j')$ that hold *inclusion* relationship and matched labels with query edge $e = (r_i, r_j)$, it is very possible that the area ratios of (r_i, r_j) and of (r_k', r_m') are significantly different provoking that "wrong" edges are returned. Hence, we compute the area-ratios for all the paired regions linked by edges and those pairs that hold significant difference (more than 20%) with the query pair are removed.

When measuring the structural similarity, the fact is that the key-regions with bigger size generally play more important role than the smaller ones. On the other hand, the document dendrogram tree $T(R, E, l)$ normally contains more edges that consisted of small key-regions than the big key-region edges. Thus most votes might be made by the small key-region edges for measuring the similarity. Hence, we apply key-region filtering process while the probability for each key-region to be kept is $p(r) \propto Area(r)$ where $Area(r)$ denotes the size of key-region r. If the given key-region is filtered out, then its immediate child is linked with its immediate parent. Hence, many edges that consisted of small key-regions are ignored during querying process. This filtering strategy is also used to reduce the retrieval time as less pairs are generated. Besides, some details of the query is kept.

6.2 Experiments

We have tested the proposed framework on the invoice dataset. The experiments were performed in two parts: 1) full page based document image retrieval to experimentally verify the optimal parameters for the proposed framework. Meanwhile, we will illustrate that the proposed method can be applied for full page retrieval yielding the state of the art results. 2) focused image retrieval demonstrating that the proposed method can adaptively deal with both *Structure-focused* querying and *Exact* matching scenario by tuning the importance of the two employed visual features. The top-10 retrieval results of the three different types of queries are illustrated in Figs. 6.5, 6.6, 6.7.

In this paper, n_geom and n_des are employed to denote the number of centroids for geometrical and content feature respectively. The Mean Average Precision (MAP) method is employed to evaluate the performance of our proposed and state of the art methods.

6.2.1 Full Page Document Image Retrieval

We demonstrate that our method can easily applied for retrieving full-page documents as well. In this section, we use BoW approaches, including the standard BoW, Spatial Pyramidal BoW and pair-wise BoW that described in Chapter 5, as baseline methods to compare the performance with the proposed one. The comparison between BoW and Pyramidal BoW demonstrates the benefits of taking the structural information into account. Besides, the performance difference between the proposed method and Spatial Pyramidal BoW (noted as PyramBoW) is also discussed to illustrate the

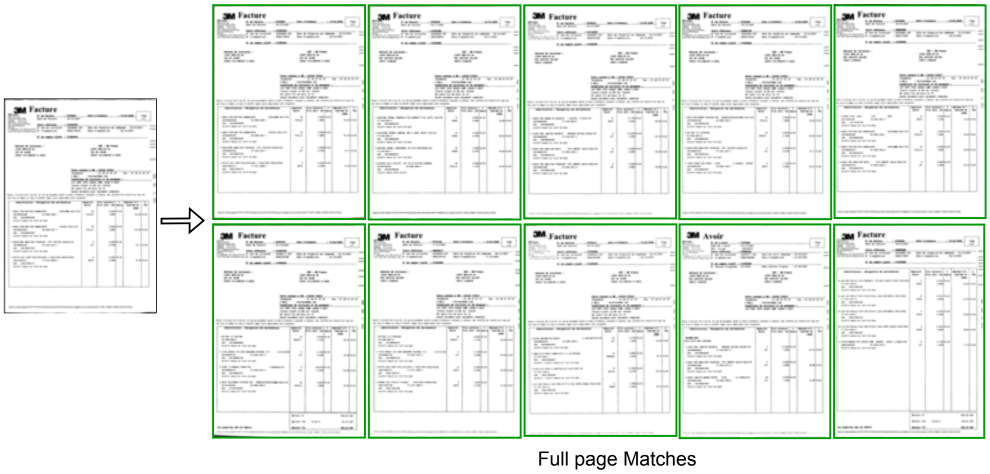


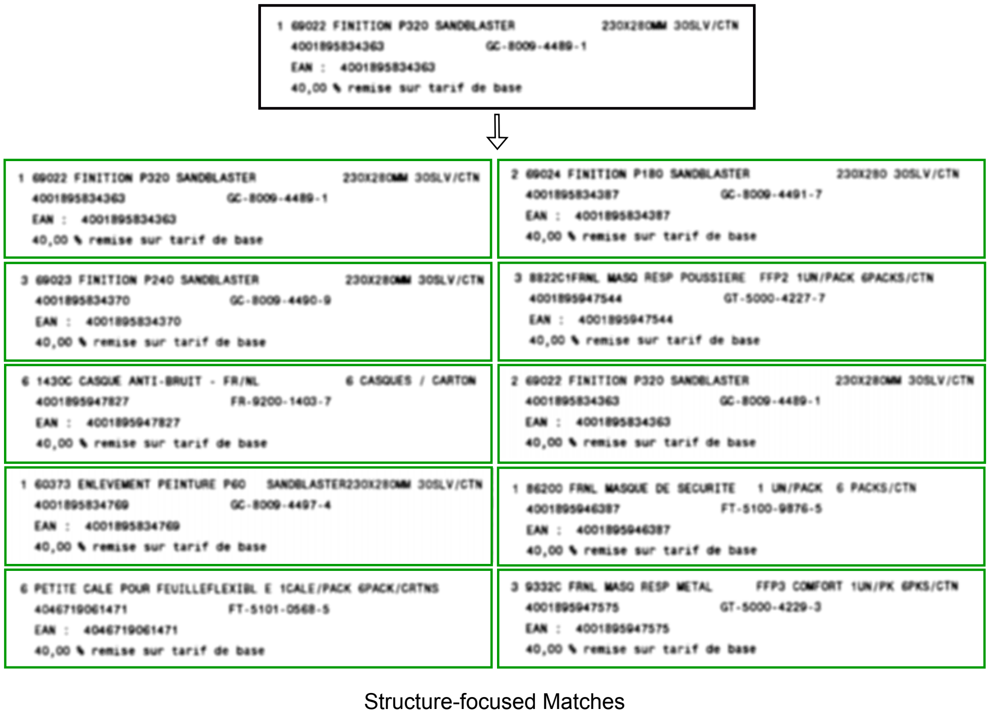
Figure 6.5: Illustration of the Top10 results of the proposed framework for full-page retrieval.

improvement of our proposed explicit structural (*inclusion*) strategy over implicit structure encoded in PyramBoW through location of local features.

For full page document retrieval, the images from the same category are generally considered equivalent. In the case of invoices employed in our experiments, the ground truth on full page structural retrieval is defined by their providers since the images from the same provider share the same template. Following leave-one-out strategy, 4109 queries are performed and, for each query, the structural similar images are retrieved from 4108 database images. Mean Average Precision (MAP) evaluation measure is applied here to evaluate the overall performance of the proposed framework.

Since the labels assigned by the quantization process are the only visual description of DTMSER key-regions, the previously computed codebook plays a very important role in the retrieval performance. Consequently, we validate the factors that could significantly affect the codebook quality which are the number of centroids for geometrical feature ($n\_geom = \{5, 10, 15, 20, 25\}$) and for content feature ($n\_des = \{50, 100, 150, 200, 250\}$) as well as the content feature description strategy (RL, ISFT, or HOG).

For the proposed structural retrieval methods based on explicit *inclusion* relations within key-region pairs, the performance improvement is generally observed when n_geom and n_des are set to a big value as shown in Fig. 6.8(c). It is because the assigned labels are not discriminative enough when clustering all key-regions into very small amount of centroids ($n\_geom = 5$ and $n\_des = 50$). However, when the number of centroids for geometrical and content feature are big enough ($n\_geom >= 15$ and $n\_des >= 150$), further significant improvement could not be achieved anymore provoking that the performance reach the ceiling limitation. Hence, considering the applicability for other situations, we select a little bigger centroids size that is $n\_geom = 20$ and $n\_des = 250$ as the optimal option.



1 69022 FINITION P320 SANDBLASTER 230X280MM 30SLV/CTN
4001895834363 GC-8009-4489-1
EAN : 4001895834363
40,00 % remise sur tarif de base

1 69022 FINITION P320 SANDBLASTER 230X280MM 30SLV/CTN
4001895834363 GC-8009-4489-1
EAN : 4001895834363
40,00 % remise sur tarif de base

2 69024 FINITION P180 SANDBLASTER 230X280 30SLV/CTN
4001895834387 GC-8009-4491-7
EAN : 4001895834387
40,00 % remise sur tarif de base

3 69023 FINITION P240 SANDBLASTER 230X280MM 30SLV/CTN
4001895834370 GC-8009-4490-9
EAN : 4001895834370
40,00 % remise sur tarif de base

3 8822C1FRNL MASQ RESP POUSSIERE FFP2 1UN/PACK 6PACKS/CTN
4001895947544 GT-5000-4227-7
EAN : 4001895947544
40,00 % remise sur tarif de base

6 1430C CASQUE ANTI-BRUIT - FR/NL 6 CASQUES / CARTON
4001895947827 FR-9200-1403-7
EAN : 4001895947827
40,00 % remise sur tarif de base

2 69022 FINITION P320 SANDBLASTER 230X280MM 30SLV/CTN
4001895834363 GC-8009-4489-1
EAN : 4001895834363
40,00 % remise sur tarif de base

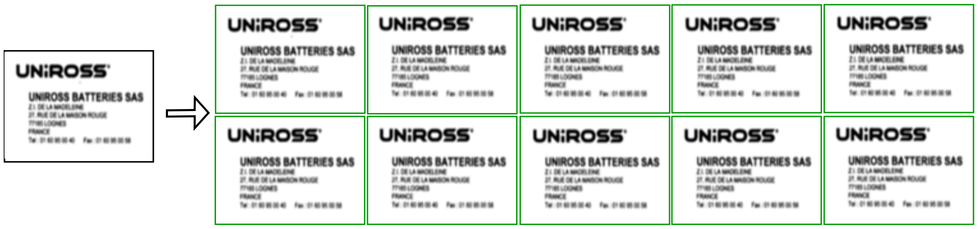
1 60373 ENLEVEMENT PEINTURE P60 SANDBLASTER230X280MM 30SLV/CTN
4001895834769 GC-8009-4497-4
EAN : 4001895834769
40,00 % remise sur tarif de base

1 86200 FRNL MASQUE DE SECURITE 1 UN/PACK 6 PACKS/CTN
4001895946387 FT-5100-9876-5
EAN : 4001895946387
40,00 % remise sur tarif de base

6 PETITE CALE POUR FEUILLEFLEXIBL E 1CALE/PACK 6PACK/CRTNS
4046719061471 FT-5101-0568-5
EAN : 4046719061471
40,00 % remise sur tarif de base

3 9332C FRNL MASQ RESP METAL FFP3 CONFORT 1UN/PK 6PKS/CTN
4001895947575 GT-5000-4229-3
EAN : 4001895947575
40,00 % remise sur tarif de base

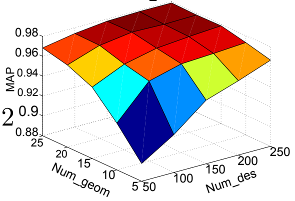
Figure 6.6: Illustration of the Top10 results of the proposed framework for structure-focused retrieval.

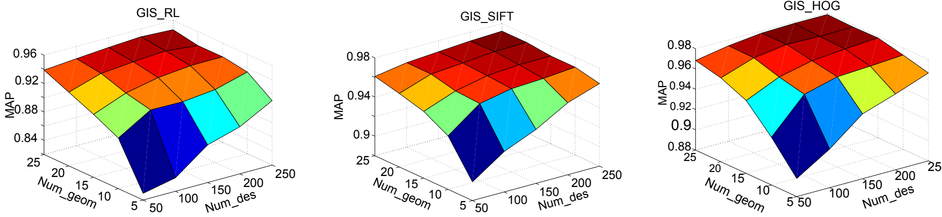


UNiROSS
UNIROSS BATTERIES SAS
Exact Matches

Figure 6.7: Illustration of the Top10 results of the proposed framework for exact retrieval.

To fairly compare the proposed method over the three considered BoW-like methods, we apply their corresponding optimal parameters that have been validated in Section 5.3.1. The detailed results of the validation on the baseline methods could be found at Figure 5.7, 5.8 and 5.9 respectively. As shown in Figure 6.8, comparing the proposed spatial indexing based framework with the BoW-like approaches, very similar behavior could be observed when tuning the number of centroids for the visual features: increasing the codebook size for either geometrical or content feature, re-





a) b) c)

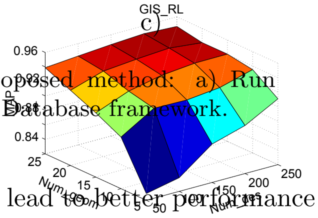


Figure 6.8: Clustering parameter validation of the proposed method: a) Run Length, b) SIFT and c) HOG descriptor based on Spatial Database framework.

gardless of description methods (RL, SIFT, HOG), would lead to better performance because larger codebook result to more discriminative power of relative feature. Besides, the ceiling point is also observed when $n\_geom > 15$ and $n\_des > 150$. Hence, we experimentally select $n\_geom = 20$ and $n\_des = 250$ as the optimal choice for the number of clustering centroids. Besides, we should point out that the increasing number of clustering centroids leads to higher dimension BoW and spatial BoW representation while the proposed method would require less retrieval time due to the implementation details of the spatial database.

Concerning the three considered description algorithm for content features,in Section 5.3.1, it was demonstrated that RL generally performs worse that HOG or SIFT. We think it is because the RL descriptor only encode the information about the number of object pixels which is less discriminative than the gradient information that applied in SIFT and HOG description. As shown in Figure 6.8, HOG and SIFT descriptors serve much better for the spatial indexing based retrieval framework than RL does.

In summary, the performance variations when tuning the parameters are highly consistent on different retrieval frameworks (BoW, Spatial Pyramidal BoW, Pair-wise BoW and the spatial indexing based retrieval). Such consistency on performance change make our conclusion about parameter validation more solid.

Table 6.2: MAP performance of descriptors and frameworks($n\_geom = 20$ and $n\_des = 250$)

	BoW	BoW_Pyram	Proposed	Pair-wise BoW
RL	0.9242	0.9430	0.9404	**0.9559**
SIFT	0.9451	0.9631	0.9723	**0.9802**
HOG	0.9494	0.9686	0.9780	**0.9816**

Table 6.2 shows the MAP performance of BoW, Pyramidal BoW and the proposed framework based on SIFT descriptors. Benefiting from the spatial/structural information, Pyramidal BoW achieved better performance than BoW framework (generally 2 percent more of MAP) illustrating the advantage of taking implicit structural (spatial) information into account for document image retrieval. Comparing with Spatial

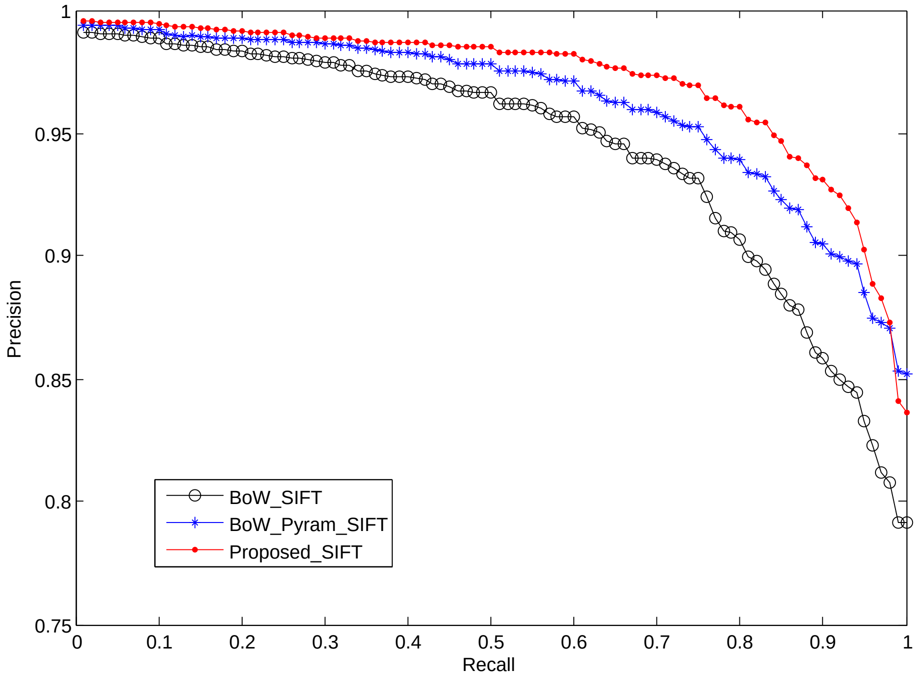


Figure 6.9: Precision Recall curve of SIFT descriptor based on BoW, Pyramidal BoW and the proposed method.

Pyramidal BoW, the proposed framework outperforms Pyramidal BoW indicating the advantage of the explicit *inclusion* structure over the implicit rough location structure. Figure 6.9 demonstrates the improvement of our proposed method over the other two frameworks through precision-recall curves.

However, comparing with Pair-wise BoW which also encode *inclusion* structural information, the spatial indexing based retrieval method generally achieve 1 percent lower MAP. It is because, when searching key-region pair $e = (r_i, r_j)$, the spatial database tests all the regions in the whole sub-branch of r_i while the Pair-wise BoW method only return check the immediate child key-regions of r_i. Such 'enhanced' spatial indexing strategy actually leads to 'noisy' results such as parent-grandchild key-region pairs while the query key-region pairs is in parent-child manner (Pair-wise BoW only return parent-child pairs). The consistency checking on area ratio that discussed in Section 6.1.3 is applied to limit such noise. On the other hand, such 'enhanced' spatial indexing strategy allows to directly query the structural relations between key-region pairs and thus makes the focused retrieval feasible while Pair-wise BoW framework could only search document at full page level.

In summary, for full page document image retrieval, 20 geometrical and 250 content feature centroids are validated to be the optimal parameters in practice. The experimental results demonstrated that the proposed methods achieve around 3 percent improvement over BoW and 1 percent over Pyramidal BoW. Besides, more importantly, the proposed method could easily be adapted to the scenario that allows the user to query part of document image and locate either *Exact* matches or *Structure-focused* matches.

6.2.2 Focused Retrieval

We show the performance of the proposed framework on focused retrieval based on SIFT description. A bounding box is expected to be obtained from the user highlighting the most interested zone, and only the key-regions that lie within the specified bounding box are employed for generating key-region pairs. Based on the matching result returned from spatial database through pair-wise querying, the consistency between the query region pairs and matched pairs is calculated by employing RANSAC algorithm. The correspondences are located by affine transforming the query boundingbox into the target images according to estimated homography matrices while a re-ranked process is performed based on the number of inliers.

To evaluate the performance of the proposed method, we define 20 part-based queries that could be divided into two groups (10 queries for each group):

- *Structure-focused* queries– the queries that aiming at retrieving all the structural similar parts while the content inside may changes from one to another. For example, in our experiment, the shopping records from invoices of the same provider may varies over item name, quantity and price while structural similarity is still kept.

- *Exact* queries– the queries that looking for the exact matches where both content and structure of the target image parts are concerned. In the experiment, we take the invoice headline as query and search the invoices that contain the same structure and the same content.

Since one single invoice may contains many such instances, we perform an iterative RANSAC process on each image while all the matched key-region pairs in the previous iteration are not considered in the following iteration. This allows us to retrieve as many matched parts as possible only if the amount of inliers is bigger than 5.

Besides, we manually create ground truth through specifying all the bounding box(es) that corresponding to *Structure-focused* queries or *Exact* queries. During query time, for each retrieved match, the query bounding box specified by the user highlighting the interested zone is affine transformed into the target image according to the corresponding matching homography matrices returned by RANSAC and the overlap area ratio criteria between transformed bounding box and ground truth bounding box is employed to determine if the retrieved matches are true or not. Inspired by the protocol of PASCAL [125], we set the threshold for this criteria to 50%.

In this section, Q1-Q10 are employed to represent the queries that correspond to *Structure-focused* queries while Q11-Q20 correspond to *Exact* queries. The "Aver." means the average performance of the respective query type Mean Average Precision (MAP) evaluation method is employed. For each query, we firstly test 8 different configurations on $\{n\_geom, n\_des\}$ (number of centroids on geometrical and content feature respectively): {100,1}, {20,1}, {20,50}, {20,200} etc. demonstrating the performance changes when tuning the discriminative power of geometrical and content features.

Table 6.3: Detailed performance on *Structure-focused* queries

	{1,50}	{1,200}	{20,1}	{20,50}	{20,200}	{100,1}	{100,50}	{100,200}
Q1	0.7228	0.6545	0.5732	0.6319	0.5049	0.6808	0.4292	0.3186
Q2	0.5111	0.5556	0.4444	0.1556	0.1852	0.2370	0.1185	0.0593
Q3	0.3397	0.3301	0.4641	0.1435	0.1005	0.4450	0.0287	0.0383
Q4	0.6036	0.6941	0.4788	0.7272	0.4973	0.6299	0.4853	0.3636
Q5	0.2875	0.2375	0.2708	0.1833	0.0208	0.4375	0.0542	0.0083
Q6	0.5102	0.4574	0.8281	0.5376	0.3441	0.9430	0.3763	0.2581
Q7	0.6887	0.6698	0.6887	0.6321	0.5000	0.6887	0.4623	0.3491
Q8	0.7340	0.6596	0.8298	0.6702	0.7872	0.9894	0.6809	0.5745
Q9	0.3726	0.2594	0.5646	0.2972	0.0566	0.8491	0.1792	0.0283
Q10	0.5000	0.8295	0.4487	0.6410	0.3846	0.8974	0.2692	0.1538
Aver.	0.5270	0.5348	0.5591	0.4620	0.3381	**0.6798**	0.3084	0.2152

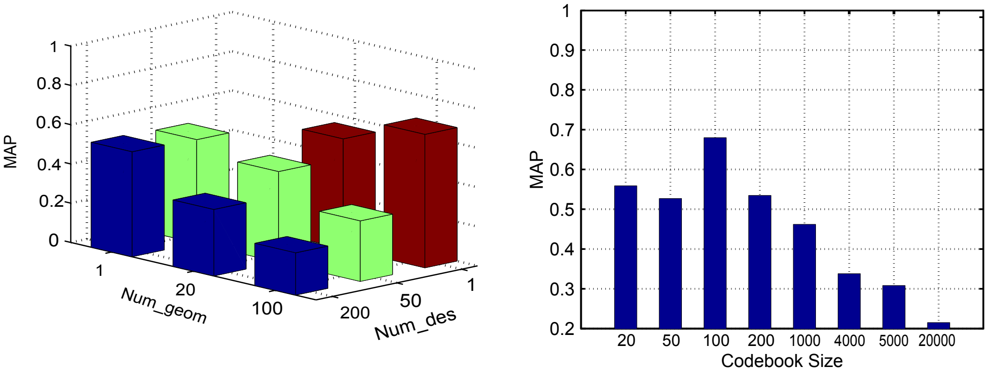


Figure 6.10: MAP performance for *Structure-focused* queries in terms of clustering configuration and codebook size respectively

Table 6.3 demonstrates the detailed retrieval results of *Structure-focused* queries that are looking for their structural similar counterpart while the change of the conveyed content is acceptable. The average performance of these ten queries is plotted in Fig. 6.10 in terms of n_geom, n_des configuration (left) and codebook size (right). When the content feature is not employed ($n\_des = 1$, correspond to the back row or the left plot), a significant improvement is observed by increasing the discriminating power of geometrical feature which allows content variation ($\{20, 1\} \rightarrow \{100, 1\}$). However, when we only use the content feature ($n\_des = 1$), the performance does not change much when increasing n_des from 50 to 200. Besides, from the left plot, the performance drop sharply when the codebook size is bigger than 1000 provoking that the codewords employed are too discriminative. The higher MAP of $\{20, 1\}$ over

$\{1, 200\}$ indicates that 20 geometrical codewords serve even better than 200 content codewords for the *Structure-focused* queries. Consequently, when the variation on content is allowed in the matches, it would be better to concentrate on geometrical features.

Table 6.4: Detailed performance on *Exact* queries

	{1,50}	{1,200}	{20,1}	{20,50}	{20,200}	{100,1}	{100,50}	{100,200}
Q11	0.8485	0.9899	0.5960	1.0000	1.0000	1.0000	1.0000	1.0000
Q12	0.4586	1.0000	0.6165	1.0000	1.0000	1.0000	1.0000	0.9925
Q13	1.0000	0.9828	0.5000	1.0000	1.0000	1.0000	0.9828	0.9828
Q14	0.9684	0.9994	0.6126	0.8603	0.8689	0.7242	0.9050	0.8728
Q15	1.0000	1.0000	0.9855	1.0000	1.0000	1.0000	1.0000	1.0000
Q16	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000
Q17	1.0000	1.0000	0.9474	1.0000	1.0000	1.0000	1.0000	1.0000
Q18	1.0000	1.0000	0.8350	1.0000	1.0000	1.0000	1.0000	1.0000
Q19	0.7619	0.9978	0.8571	0.9959	0.8571	1.0000	0.8095	0.8571
Q20	1.0000	0.9828	0.7748	0.9825	0.9446	0.5161	0.7931	0.7414
Aver.	0.9037	**0.9953**	0.7725	0.9839	0.9671	0.9240	0.9490	0.9447

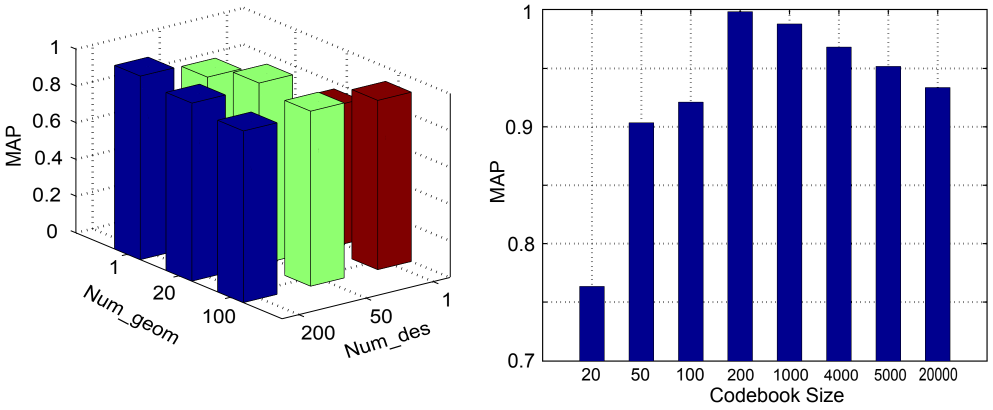


Figure 6.11: MAP performance for *Exact* queries in terms of clustering configuration and codebook size respectively.

On the other hand, for the queries that searching for *Exact* content-based matches which hold less in-class variation (especially on the content), the proposed system achieves much higher MAP comparing with the *Structure-focused* queries. With only geometrical feature ($n\_des = 1$), better performance is achieved when increasing the number of geometrical codewords from 20 to 100. Such improvement could also be observed if the number of content codewords is increased from 50 to 200 when geometrical feature is not employed ($n\_geom = 1$). However, as shown Fig. 6.11, when the codebook size is bigger than 1000, increasing the discriminative power of either geometrical or content feature would lead to performance drop provoking that

Table 6.5: Performance on part-based queries (codebook size fixed)

	{100,1}	{25,4}	{10,10}	{4,25}	{1,100}
Structure-focused	0.6771	**0.6805**	0.6062	0.5780	0.5519
Exact	0.9253	0.9698	**0.9938**	0.9891	0.9921

the codewords we used might be too discriminative.

As discussed above, for either *Structure-focused* or *Exact* queries, the assigned labels encoding the content and geometrical features easily become to be too discriminative. Hence, in Table 6.5, we further test the performance change when tuning the importance of the two types of features while overall information content is roughly limited in an proper range where over-discriminative problem rare occur.

We have fixed the overall codebook size to roughly limit the amount of overall discriminative power of the employed codewords. Afterwards, we have studied the behavior of the proposed system when tuning the importance of geometrical and content codewords. As shown in Table 6.5, for *Structure-focused* queries, the improved performance of utilizing four content codewords ($\{n\_geom = 25, n\_des = 4\}$) over the configuration that only use geometrical information ($\{n\_geom = 100, n\_des = 1\}$) indicates the advantage of taking the content feature into account. However, paying more attention on content feature will lead to less discriminative power on geometrical feature and thus would generally result in worse retrieval performance. On the other hand, for *Exact* queries, increasing the importance of content feature normally results in improved retrieval performance while $\{n\_geom = 10, n\_des = 10\}$ could be considered as the ceiling configuration.

In summary, we demonstrated that the proposed system can adaptively return *Structure-focused* matches and *Exact* matches by tuning the discriminative power of geometrical and content features. The experiments demonstrate that concentrating on the geometrical feature could generally enhance *Structure-focused* retrieval performance. The reason is geometrical features are more tolerable for content change and thus structure feature (*inclusion*) can play the key role in retrieval process. On the other hand, for *Exact* queries that searching for the matches that hold similarity on both structure and content, adding more information on either geometrical or content feature would result in more discriminative power of key-region descriptions. such enhanced visual features in turn lead to more discriminative power and thus better retrieval performance.

6.3 Conclusion

In this paper, we have proposed a new framework for structural document image retrieval that allows to query structure elements such as key-region pairs, triplets or group of key-regions linked with various structural relationships (*inclusion*, *intersection*, *top/left of* or even *within specific distance*, etc.) while *inclusion* relation within key-regions is employed in this paper. By tuning the discriminative power of the two types of feature description, we have demonstrated that the proposed system could adaptively retrieving the *Structure-focused* and *Exact* matched parts. For example, when searching for the exact match that holds similarity on both structure and content, one would benefit from enhancing the importance of content feature. On contrast, focusing on geometrical feature generally results in better retrieval performance when the user only expects the counterparts that only hold similarity on structure while the change of contained content is allowed. Besides, we also applied the proposed method for full-page image retrieval and achieve better results comparing with the baseline methods.

Chapter 7

Spatial Verification

We introduced a generic framework in Chapter 6 that is capable to retrieve the document images by structural and visual similarities for both full page or focused-based retrieval scenarios. Key-region pairs with *inclusion* structural relations are employed to retrieve their counterparts from the database while the number of the matched key-region pairs is applied to measure the similarity between query and the target images (or image parts in the case of focus-based retrieval scenario). RANSAC algorithm is then performed to check the spatial consistency between query key-region pairs and the matched pairs in the target image.

The RANSAC algorithm have achieved great success for various applications especially when the exact matches are expected (e.g. image stitching, logo retrieval etc.) However, it is observed that RANSAC becomes to be too rigid when searching for the matches which are structurally similar with query while the conveyed text might change (e.g. address block, shopping items etc.) Hence, in this chapter, we will introduce a geometry verification method as an alternative option of the conventional RANSAC algorithm to make sure the final matches follow the same spatial transformation.

7.1 Spatial Verification

Spatial Verification process is usually employed to estimate the transformation relations to filter out the 'bad' matches (outliers) that are inconsistent with the estimated transformation. The transformation relations generally can be formulated as a transformation matrix as follows (also known as homography matrix for many applications).

$$\begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = \begin{bmatrix} \alpha\cos\theta & -\alpha\sin\theta & \alpha t_x \\ \alpha\sin\theta & \alpha\cos\theta & \alpha t_y \\ 0 & 0 & 1 \end{bmatrix} \times \begin{bmatrix} x\prime \\ y\prime \\ 1 \end{bmatrix} \tag{7.1}$$

where α, θ and (t_x, t_y) represent the scale change, rotation and translate respectively. Since the invoice images are well scanned in a flat plane, shearing seldom is observed between images. Consequently, we only consider the three types of transformation.

Spatial verification process takes $XY = \{(xy_1, xy_1\prime), (xy_2, xy_2\prime), \ldots, (xy_n, xy_n\prime)\}$ as inputs whereas $(xy_1, xy_2, \ldots, xy_n)$ and $(xy_1\prime, xy_2\prime, \ldots, xy_n\prime)$ represent the location of the matched points for query and target image respectively and n denotes the number of the matched points. Then it employs such observations to estimate the parameters for the transformation matrix shown in Equation 7.1. Various of methods on spatial verification, such as RANSAC algorithm and geometrical verification methods, have been introduced for many applications to refine the corresponding matches and further improve the matched instances.

7.1.1 RANSAC algorithm

RANSAC algorithm is an iterative process for estimating the transformation matrix. It random sample specific number of matched points as a initial census/pool. Afterwards, an iterative calculation performed on the two following progress: 1)estimate the value of parameter α, θ and (t_x, t_y) for transformation matrix using the current census; 2) update the census by adding more points from location pairs XY if they are consistent with the estimated transformation and remove the ones that are not consistent with the new transformation from the previous census. The two step discussed above are repeated until the algorithm converge (the census and transformation is 'updating' anymore) or the given maximum iteration number is reached.

Generally, RANSAC algorithm makes an initial assumption on transformation and add new 'inliers' from all the observation for such assumption and afterwards update the new assumption according to the updated group of 'inliers'. Hence, through the two iterative progress, RANSAC is capable to arrive maxima point in parameter space from the initial assumption through iterative 'minor' revision. However, when large portion of outliers appears in the given location pairs, RANSAC algorithm might fails to obtain the right estimation for transformation. The reason might be that RANSAC check the spatial consistency for all the matches in a global manner and thus the right 'answer' would easily hidden from large portion of noise.

7.1.2 Geometrical Verification

Geometrical verification is an alternative type of methods for verifying the spatial consistency of the pairs of matched points. Differently with the global manner that employed in RANSAC algorithm, geometrical verification locally check the geometrical consistency among small amount of matched points. For example in [14], geometric verification process is employed to compute the consistency of the matched SURF key-points for logo retrieval scenario.

As shown in Figure 7.1, triangular geometrical verification compares the corresponding triangles from query and target image and determine if their transformation is consistent with the ones estimated through other triangles. Usually, when there are only small amount of points, all combinations of 3 matched points are employed to generate the triangles. The transformation that largest amount of triangles demonstrate consistent is returned as the final estimation of transformation between query and target image.

However, the computational cost of such triangular verification process is $O(n^3)$

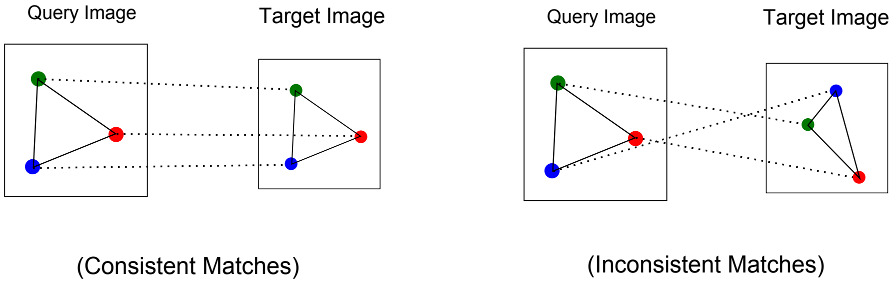
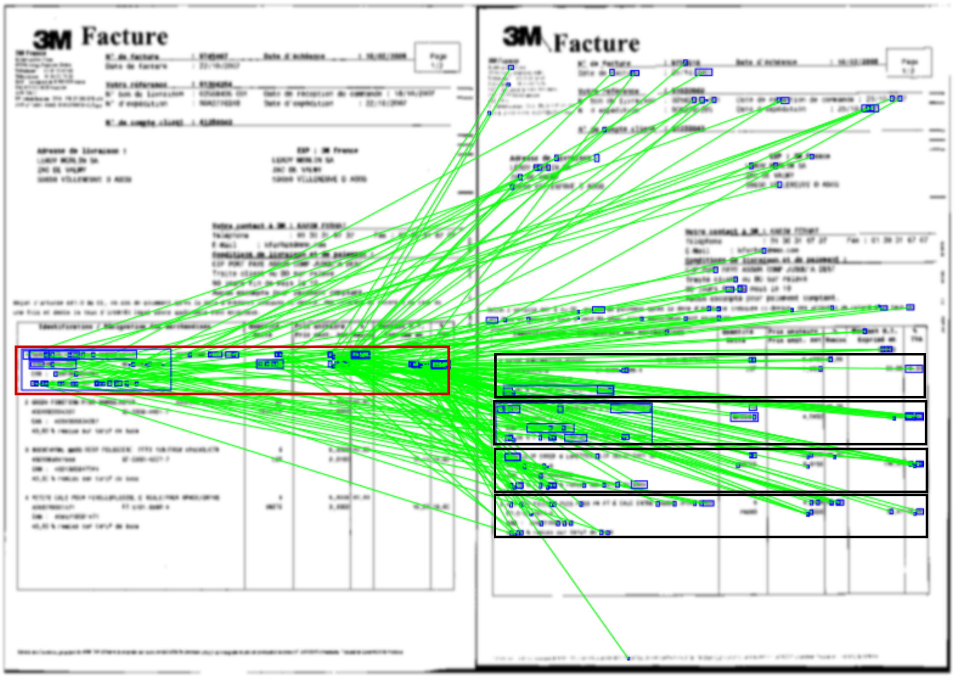


Figure 7.1: Geometrical verification by checking the consistency between triangles.

where n is the number of the matched points. Hence, it become to be infeasible when n grows. The common strategies to low down its computation time are: 1) sample a subset with small size from all the matched points. 2) find a reference matched point that will surely appear in all the target images and only compute the triangles related with the reference point, thus the computational cost boiled down to $O(n^2)$. For example, the number of effective matched points are limited to 25 in [14] while other matches with large distance are not employed. However, as shown in Figure 7.2, we seek for retrieving all the similar image parts from the dataset while the multi instances are expected from one target image. Hence, it is not feasible in our case to limit the number of matches into a proper range. Besides, as explained in Section 6.2.2, apart from the exact matches, we also aim at searching the structure-focused counterparts in a more flexible manner where the content change is allowed. Consequently, strategy 2) also do not serve our situation since there might not exist stable reference point that appears in all the counterparts.

Line verification is a cheaper option for geometrical verification. Similar with triangular version, it takes paired X-Y location of the matched points and estimate the transformations according to certain local geometrical relations. The difference between line verification and triangle verification is that the former method compute the estimation based on lines between points. As shown in Figure 7.3, the triangle that employed in triangular verification boils down to three independent lines which are further used to estimate three separate transformations ($\{\alpha_i, \theta_i, t_{xi}, t_{yi}, i = 1, 2, 3\}$) for the matched points. Line verification computes all the combinations of two matched points each of which corresponds one estimation on transformation. At the end , the transformations are determined by finding the ones that have large amount of supporting lines (observations).

Comparing with triangle verification, line verification is less expensive since its computation complexity boils down to $O(n^2)$. Besides, line verification is more flexible than triangle one. As shown in Figure 7.3 (the inconsistent matching situation), even though matched point 1 is an outlier and the related lines lead to wrong estimation, one correct transformation still can be computed from the line connecting point 2 and 3 (dotted mapping line in blue).



(Query Image) (Target Image)

Figure 7.2: The example of matched points (key-region pairs in our case). The blue bounding boxes correspond the key-region pairs while the big red and black ones indicate the focused query part and its expected matches respectively. Only 20% of the overall matches are shown here for a nice visual.

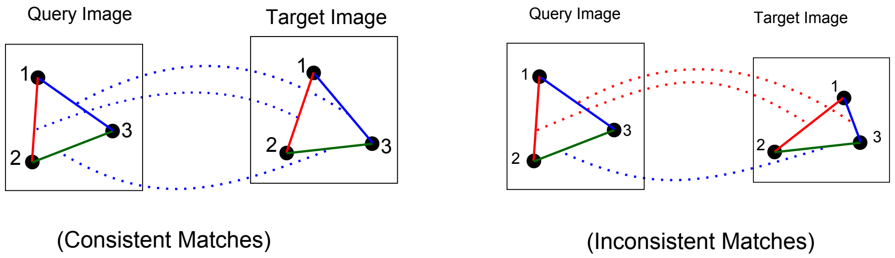


Figure 7.3: Geometrical verification by check the relations between lines and estimate the transformation for the matched points. The dotted lines show the mapping relation of lines. The blue dotted lines correspond to correct transformation estimation and the red dotted line indicate the wrong ones.

7.1.3 Proposed Line Verification

Line verification hold $O(n^2)$ computation complexity that is much cheaper than triangle verification. However, it is still unaffordable expensive when multi counterparts are expected. As shown in Figure 7.2, most of the matched points can not eventually removed before performing line verification. Hence it is infeasible in our situation to reduce the computation time through lowing down the number of matched points n. In [126], its computation complexity is reduced to $O(n)$ through firstly find a reference point and only employ the lines that related to the reference point. However, we seek for a generic retrieval framework that is not limited to only search the exact matches only. Hence, searching in such flexible manner does not allow to firstly fix a reference point because no firm matched points can be guaranteed.

In this section, we will introduce an variation of line verification method. As shown in Figure 7.2, when multi instances are expected in the target image, there are large number of correct matched points that will be positively contribute for line verification process. However, it not necessary to employ all the combination of points to generate the lines. For instance, the line that links one point from the first bounding box in the target image (see Figure 7.2) and another point from other bounding boxes will not lead to expected transformation estimation. Consequently, we propose to performline verification in a two step manner as follows.

- **Step 1**: we firstly estimate the tentative bounding boxes (shown in Figure 7.4 as the blue rectangles) to divide the matched points into several groups according their locations. To obtain those bounding boxes, each match is employed to compute the transformation parameters while the scale $\alpha = Area_i/Area_{i'}$ and the rotation $\theta = Orient_i - Orient_{i'}$ where $Area_i$ and $Orient_i$ represent the area and the orientation of the parent regions of the ith matched key-region pairs in the query and $Area_{i'}$ and $Orient_{i'}$ correspond to the counterparts in target image. The translation is determined by the location by the two corresponding matched points. The estimated transformations that at least fit $Thre1$ points are selected as tentative bounding boxes($Thre1$ is experimentally set to 10.)
- **Step 2**: geometrical verification is employed to precisely check the spatial consistency among the points insides each tentative bounding boxes. Those bounding boxes is updated through finding the one that fit largest number of matched lines (inliers). Similarly with step 1, we set a threshold $Thre2$ on number of consistent lines (inliers). When $Thre < 3$, the estimated transformation is removed due to too few inliers. Step 2 is repeated one time for the each of updated bounding boxes to refine the result.

Comparing with the conventional line verification process, the introduced two-step version is much cheaper for calculation its rough computation complexity is $O(n + k * (\frac{n}{k})^2)$ where k denotes the number of tentative bounding boxes that found in step 1. Besides, since the lines that link two matched points from different tentative bounding boxes are not taken into account, the two-step line verification method handles less number of wrong lines and thus lead to better estimation on transformation.

7.2 Experiment Results

We test the proposed line verification methods for the focused retrieval as described in Appendix A.2. 20 queries are divided into 2 groups: exact match and structure

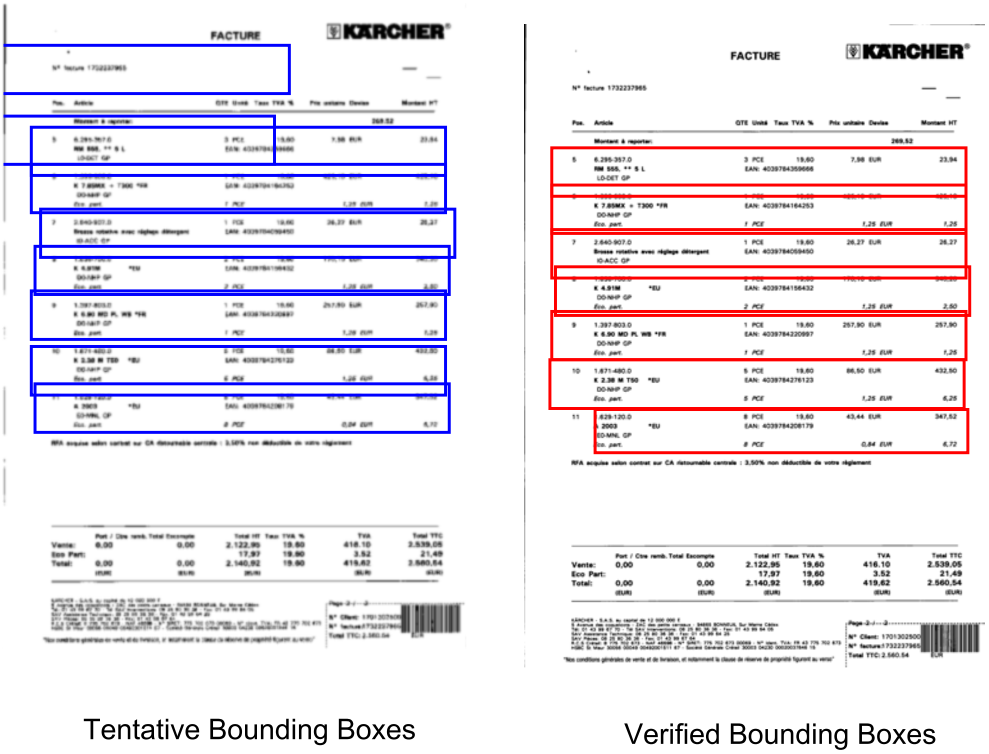


Figure 7.4: Two step line verification.

focused match (defined in Section 6.2.2). Following the experimental setup of Section 6.2.2, we employ DTMSER to generate key-pairs which is quantized according to their visual features. Afterwards, a pair-wise key-regions querying process is casted and the spatial database returns the matched pairs that demonstrate *inclusion* relation and hold the same corresponding key-region labels. However, we substitute the RANSAC algorithm with the proposed two step line verification strategy to check the spatial consistency among the matched key-regions.

As explained before, line verification is repeated two time for each tentative bounding box: first time to estimate the transformation and the second round is employed to refine the inliers and validate the estimated. Nevertheless, we also tested only perform the line verification process one time while the refinement is not applied to save around 50% time consumption.In this chapter, *LineVeri1* and *LineVeri2* stand for performing line verification for one and two times respectively while *RANSAC* corresponds to the performance achieved by RANSAC algorithm.

To compare the performance of RANSAC algorithm that presented in Section 6.2.2, we also applied line verification process to find the optimal configuration on the number of centroids for the two type of visual features. As shown in Table 7.1, the optimal parameters ($n_geom = 25n_des$) that validated for RANSAC also achieves the best performance for the line verification. Besides, similar behavior with Section 6.2.2 is

Table 7.1: Performance on focused queries whereas the structural similarity is especially focused (Q1 Q10).

	{100,1}	{25,4}	{10,10}	{4,25}	{1,100}
RANSAC	0.6771	**0.6805**	0.6062	0.5780	0.5519
LineVeri1	0.7383	**0.7970**	0.6959	0.7234	0.6110
LineVeri1	0.7400	**0.8243**	0.6887	0.7246	0.6076

observed when tuning the discriminative power of the geometrical and SIFT features.

Comparing with RANSAC, line verification generally achieves 6 14 percentage better performance despite of the employed parameter configuration. The reason for such improvement is that RANSAC compute the transformation in a rigid global manner and thus fails to retrieve the true positives when large portion of outliers appear. On contrast, line verification separately estimate the transformations by each the local lines and hence more robust on the outliers. As shown in Figure 7.2, when seeking to search multi instances in one single target image, large amount of outliers usually exists outside the expected bounding box.

Besides, we further analyze the precision and recall separately. As shown in Figure 7.5, RANSAC would generally achieve higher precision but signification lower recall than the two line verification methods. Comparing $LineVeri1$ and $LineVeri2$, representing line verification without and with refinement respectively, remarkable improvement on precision is observed when one extra verification process is applied (see the precision of the query #1, #4, #6, #9, #10). Meanwhile, As shown in 7.6, such refinement does not necessarily result in notable decrease on the corresponding recall. Regarding to the recall of the retrieved result, line verification methods consistently achieve much higher recall performance than RANSAC as shown in Figure 7.6. Such significant enhancement on recall lead to the around 14 % improvement on retrieval performance while MAP is employed as the evaluation method.

Apart of the structure focused queries, our research also aims at retrieving the exact matches. Hence, we also compare the performance of line verification methods with RANSAC. As shown in Table 7.2, the performance the retrieval with RANSAC has already reach a near-perfect state (the MAP is 0.9938). Even though, with the help of line verification which is employed to replace RANSAC, the system still make tiny but very difficult improvement (from 0.9938 to 0.9999).

Table 7.2: Performance on part-based queries (codebook size fixed)

	{100,1}	{25,4}	{10,10}	{4,25}	{1,100}
RANSAC	0.9253	0.9698	**0.9938**	0.9831	0.9921
LineVeri1	0.9828	0.9828	**0.9996**	0.9994	0.9961
LineVeri2	0.9873	0.9889	0.9998	**0.9999**	0.9964

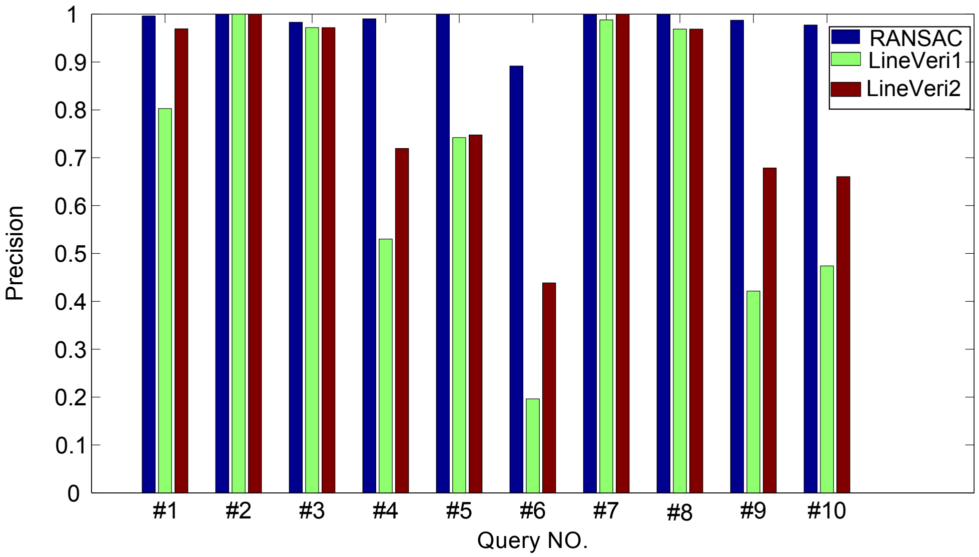


Figure 7.5: Two step line verification.

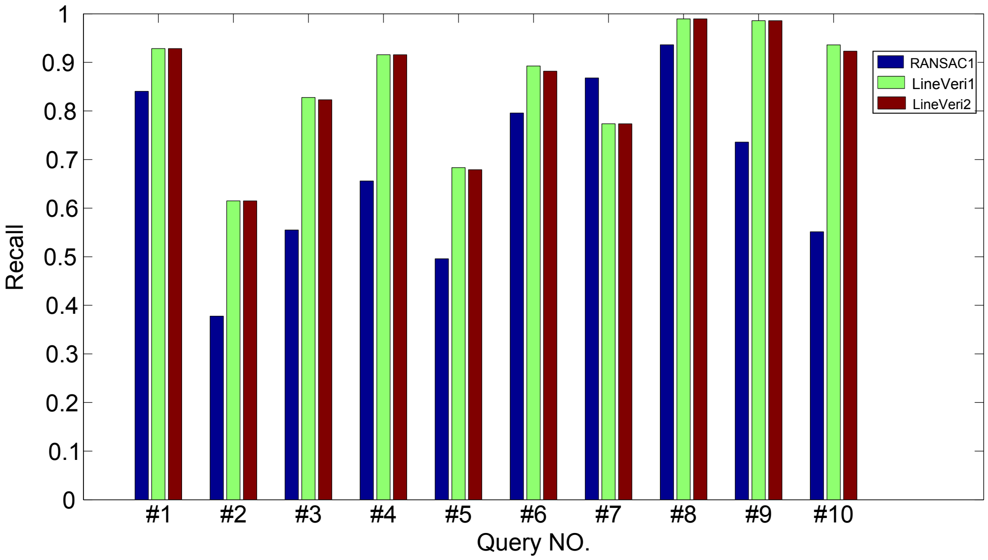


Figure 7.6: Two step line verification.

7.3 Conclusion and Future Work

In this chapter, we introduce a line verification method to compute the transformations between query and target images. We demonstrated that the RANSAC algorithm might fail to find the right estimations when seeking for multi counterparts due to large portion of outliers. Since the line verification is more flexible for outliers, it would lead to much higher recall while small loss on precision is observed comparing with RANSAC. Even though, line verification is still an optimal option when the recall matters more.

The main disadvantage of line verification is that its cost is $O(n^2)$. However, the other hand our system seeks for multi instance searching in single target images and thus lead to larger number of matched points. Even though we manage to low down its cost to $O(n + k(\frac{n}{k})^2)$ through a two step strategy, it sis still much more expensive than RANSAC. Hence, in the future, it would be nice to further boils its cost down to $O(n)$. One possible way is to find a reference points (e.g. geometrical center of the tentative bounding box) and only employed the lines related to the selected reference point.

Chapter 8

Conclusion and Future Work

In this thesis, we introduced a series of consecutive works heading to a generic framework for document image retrieval where a flexible similarity measurement based on both structural information and visual features is developed. Based on the pair-wise key-region representation carrying both visual features of key-regions and *inclusion* structural relations between them, we demonstrated that the proposed framework would serve the whole spectrum of retrieval problem, from solely structural similarity based searching to purely visual feature (e.g. SIFT) based retrieval. Besides, since the framework represent each document as a group of local key-region pairs, it allows to cast both full page queries and focused image parts where one-to-many matches is also specially considered. The path targeting to such generic framework is detailed as follows.

We firstly introduced a simple method for real-time document retrieval problem based the pyramidal structure and the corresponding density feature of the images. We demonstrated that, benefiting from the encoded pyramidal structure information, the density feature based document representation achieves remarkable performance. Besides, we also illustrated that its performance could be significantly improved by several iterations of relevance feedback process.

Based on the fact that the structure of a document is tightly linked to the distance among its elements, we proposed a distance aware version of MSER (DTMSER). We illustrated that DTMSER algorithm is able to efficiently extract multi scale semantical key-regions that roughly correspond to the structural elements of the document such as letters,words, paragraphs, etc. Meanwhile, the document structure is expressed as a dendrogram (hierarchical tree) defining how those key-regions merge to each other. We demonstrated that the DTMSER can achieve equivalent (actually slightly better) performance with state-of-the-art in a retrieval scenario by comparing much smaller amount of key-regions while the extracted structure information had not been applied.

Afterwards, we proposed a pair-wise BOW methods for full page document retrieval scenario. Each document image is represented as a list of key-pairs (correspond to the edges in its dendrogram) with *inclusion* structural relations between the related key-regions. We illustrated an efficient manner to embed such structural information into a BoW-like histogram representation by assigning the key-region pairs as pooling elements. Besides, to solve the computation complexity, the inverted file indexing strategy is employed to calculate the distance between the pair-wise histogram representations that is significant sparse. We demonstrated that, with embedded explicit *inclusion* structural information, the proposed pair-wise BoW representation achieved remarkable improvement over the conventional BoW and Spatial Pyramidal BoW methods.

We finally arrived to a generic framework for structural document image retrieval that allows to directly query structure elements such as key-region pairs, triplets or group of key-regions. We employed spatial indexing strategy facilitating to compare various structural relationships (*intersection*, *overlap*, $top/left$ of etc.) while *inclusion* relation encoded in the key-region pairs is applied in our work. By tuning the discriminative power of the two types of visual feature, we have demonstrated that the proposed system is capable to smoothly adjust the similarity between structural and visual measurement and thus allows to retrieve *Structure-focused* and *Exact* queries. We demonstrated that the proposed framework serves various retrieval scenarios including full page structural retrieval, the focused querying while the structure similarity is concerned and also the focused searching when the exact matches are expected. We showed that the generic framework achieves better retrieval results than BOW and Spatial BoW while slight performance decrease is observed comparing with the pair-wise BoW introduced in Chapter 5. Besides, we illustrated that, with the help of RANSAC to check spatial consistency, the proposed framework achieves nearly perfect precision performance while the recall is not very high for structure-focused queries even when very low threshold for number of inliers is applied.

At the end, we introduced a two step line verification, as an alternative method of RANSAC which is found to be very rigid, to compute the transformations between query and target images. We demonstrated that the RANSAC algorithm might fail to find the right estimations when seeking for multi counterparts due to large portion of 'outliers'. We discussed that the introduced line verification is more flexible for outliers because it compute the transformation from the local lines and thus other outliers will not affect the lines between the inliers. We demonstrated that the proposed line verification method would lead to much higher recall performance while small loss on precision is observed comparing with RANSAC.

Even though we showed that the proposed methods have achieved reasonable results for various retrieval scenarios, we believe that there is still a long way heading to the final solution fordocument image retrieval problems. The future directions of our research are discussed as follows.

We demonstrated that the proposed line verification is able to achieve remarkable higher recall while its computation complexity ($O(n^2)$) may retard its applications. Hence, in the future, it would nice to explore an cheaper version of line verification. One advisable direction is to employ the ker-region pair that mostly appears in the tentative bounding boxes (see Section 7.1.3) as the reference point. In this way, the computational cost of line verification process can be reduced to $O(2n)$ by only check the lines that related to the reference points. However, when the structure focused matches are expected, the found reference point may not always appear in the true positives due to the content variation. In such case, another way strategy to assign a stable reference point is to use the geometrical center of the bounding boxes.

Because the retrieval results from the proposed generic framework are usually very precise (nearly 1), we think it is worthy to employe them as positives and develop an automatic learning scheme to explore the which type of information is the user trying to search. It would be nice if the system is able to inference whether the query type, structure-focused retrieval or exact match searching, and then adaptively find the optimal configuration on the number of clustering centroids and thus lead to performance improvement. For example, if the positives retrieved by the generic system demonstrate huge variation on the content, then it is advisable to assign more importance on the structure relation by setting number of centroids to a smaller value for the visual features, especially for the SIFT features. Another direction for the learning scheme is to train a simple classifier online, e.g liner SVM, while the retrieved result from the generic framework can be employed as positives and the negative examples could be generated from the random patches. In this way, we can directly apply the trained classifier to determine if the tentative bounding boxes are a good match or not without applying the line verification which is known to be expensive. The third way for benefiting from the retrieved positives would be the its cooperation with relevance feedback process. In such case, it would be nice to explore the discriminative or symbolic key-region pairs by exploring the ones that appear in most of the positives and thus assign a higher weight to those common pairs during the voting stage.

In Chapter 6, we stated that various of spatial relationships among the stored key-regions have been indexed and ready to use. Hence, in the future, it would be worthy to incorporate other spatial relation such as *left/top* of or *within given distance* into the framework while only *incluson* relation is currently employed. The benefit of adding more spatial relations is that the retrieved results might be very discriminated in the sense of the structure of the key-regions. Consequently, it would be not necessary to employ RANSAC or any geometrical verification process to check the spatial consistency because certain type of spatial relations have already been checked when querying from the database. Another really challenging direction to improve the spatial indexing based retreival system is develop the customized indexation strategy accordingly to the need of document retrieval system. For example, in the implementation of the spatial index of the spatial database, many types of relations such as *intersection*, *overlap*, *tangency* etc. has already been incorporated. However, those spatial relations will never happen among the extracted DTMSER key-regions. Hence, the customized indexation strategy might speed up the query process while less space needed to store the indexing information.

Appendix A

Datasets

In our research, we mainly tested our retrieval system on an invoice dataset consists of 4109 images overall. Those invoices are provided by 249 companies and the documents from the same provider are generated from the same template and thus demonstrate obvious visual similarities and vice versa. The 249 invoice classes are very unbalanced while their sizes varies from 3 to 133. The statistical details of the class size are shown in Figure A.1. Hence, when the MAP measurement is employed, the performance of the retrieval system over some specific invoices maybe not precisely manifest the retrieval power of the system. That is why we use leave-one-out strategy to generate the queries images for full page retrieval. On the other hand, for the focused retrieval scenarios, we only choose the relative image parts from invoices with reasonable number of instances within their classes.

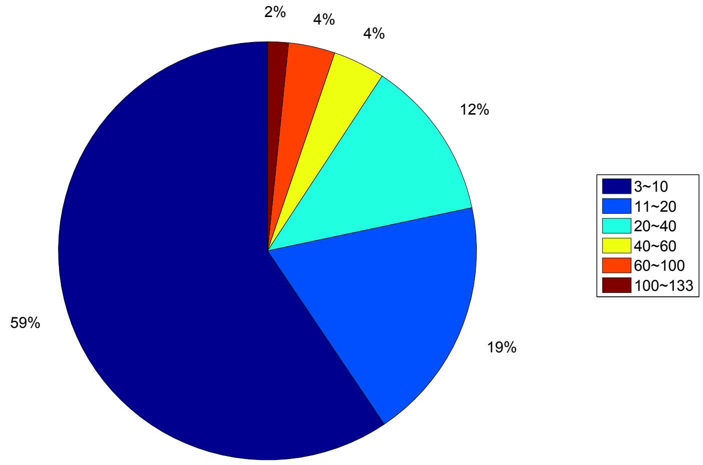


Figure A.1: The statistical distribution of the size of all the invoice classes.

A.1 Full page retrieval

The experiment on document retrieval at full page level is performed by searching the invoices that provided by the same provider with the query image. As shown in Figure, the invoices from the same provider share the same template and thus they are structurally similar. Hence, this experiment is designed to demonstrate how would it improve the retrieval performance by taking such structural consistency into account.

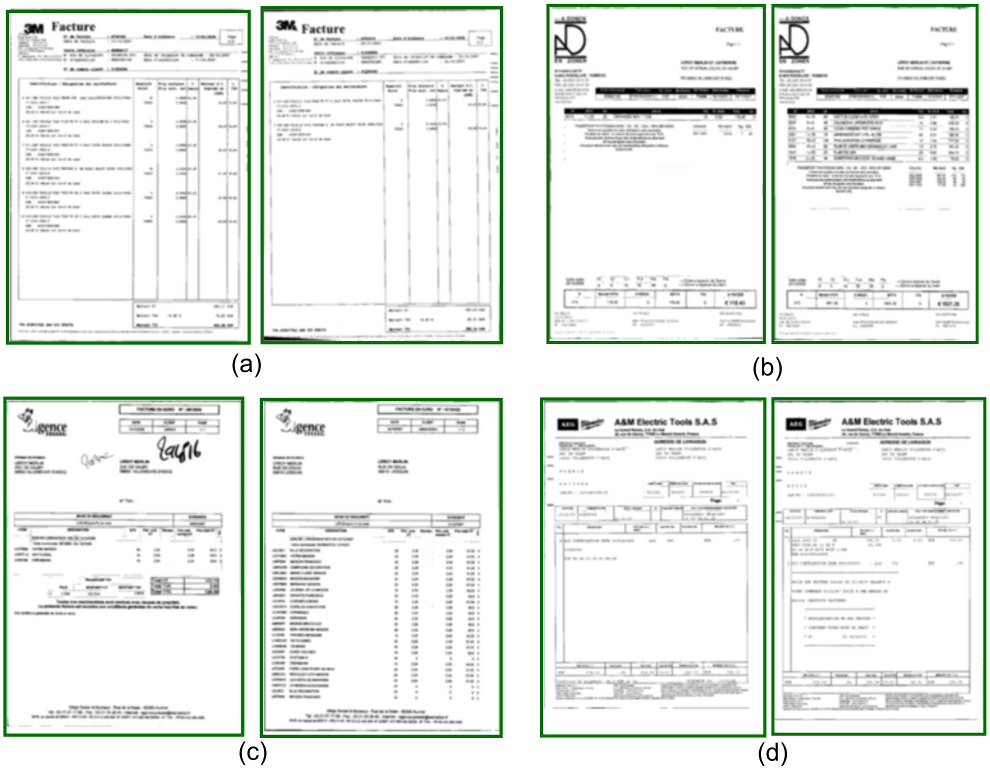
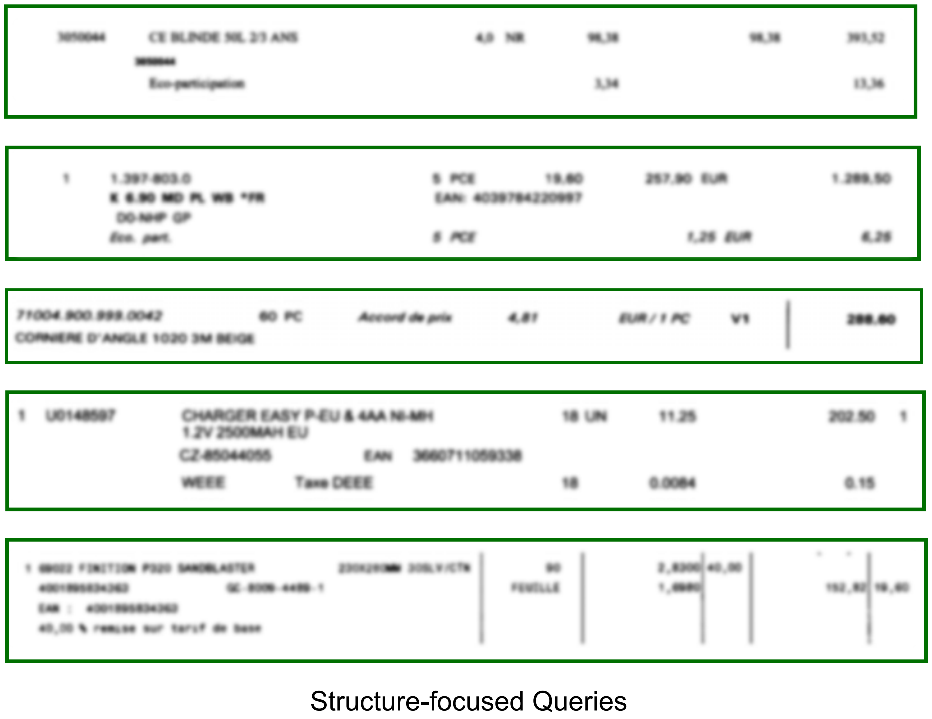

3M Facture
3M Facture
(a)
FACTURE
FACTURE
(b)
(c)
A&M Electric Tools S.A.S
A&M Electric Tools S.A.S
(d)

Figure A.2: Samples of invoice images from four providers.

As mentioned before, we employ leave-one-out strategy to generate the query and a ranked list of the remaining 4108 database images is returned by the retrieval systems. In order to determine if the retrieved images are structurally similar with the query, the ground truth for each query is defined by the provider IDs of the images. Mean Average Precision (MAP) is employed to evaluate the quality of the returned ranking list as well as the performance of the retrieval systems. Besides, the precision-recall curve, which is averaged over 4109 queries, is also employed to visually demonstrate the retrieval performance.

A.2 Focused Retrieval

Besides the full page retrieval scenario, we also test our framework on the focused queries where query images are the focused area of the documents such as logos, address blocks and shopping items. Furthermore, according to the similarity that those queries seek for, the focused queries is in turn divided into two groups *structure-focused* and *exact* queries. *Structure-focused* queries search for the counterparts that hold similarity on their structure while the variation on the carried content is allowed. On contrast, the *exact* only expect the results image parts that demonstrate high similarity on both content and content. In our research, we employ the shopping items and invoice logos to simulate the two types of queries. We defined 20 queries, 10 for each type, while the corresponding ground truth is manually tagged from the whole collection. Figure A.3 and Figure A.4 illustrate some samples of the two types queries respectively. The number of true positive for each of the queries is listed in the Table A.1 whereas "Type1", "Type2" and "T.P. Amount" represent the *structure-focused* queries, *exact* queries and the amount of the relative true positives.


3050044 CE BLINDE 50L 2/3 ANS 4,0 NR 98,38 98,38 393,52
3050044
Eco-participation 3,34 13,36

1 1.397-803.0 5 PCE 19,60 257,90 EUR 1.289,50
K 6.90 MD PL WS *FR EAN: 4039784220997
DO-NHP GP
Eco. part. 5 PCE 1,25 EUR 6,25

71004.900.999.0042 60 PC Accord de prix 4,81 EUR / 1 PC V1 288,60
CORNIERE D'ANGLE 1020 3M BEIGE

1 U0148597 CHARGER EASY P-EU & 4AA NI-MH 18 UN 11.25 202.50 1
1.2V 2500MAH EU
CZ-85044055 EAN 3660711059338
WEEE Taxe DEEE 18 0.0084 0.15

1 69022 FINITION P320 SANDBLASTER 50 2,8300 40,00 152,82 19,80
4001895834363 GE-8006-4489-1 FEUILLE 1,6980
EAN : 4001895834363
40,00 % remise sur tarif de base

Figure A.3: Samples of the structured focused queries.

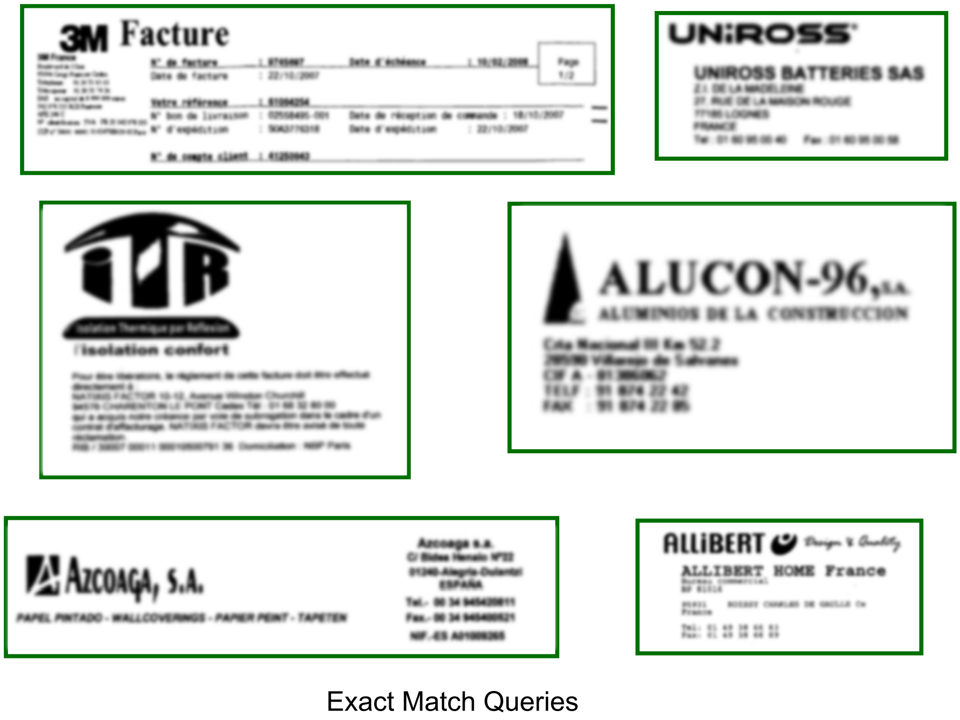

3M Facture
UNiROSS
UNIROSS BATTERIES SAS
l'isolation confort
ALUCON-96, S.A.
ALUMINIOS DE LA CONSTRUCCION
AZCOAGA, S.A.
PAPEL PINTADO - WALLCOVERINGS - PAPIER PEINT - TAPETEN
ALLIBERT Design & Quality
ALLIBERT HOME France
Exact Match Queries

Figure A.4: Samples of the queries that seek for exact matches from the collection.

Table A.1: Number of true positives of the queries for focused retrieval.

Type1	#1	#2	#3	#4	#5	#6	#7	#8	#9	#10
T.P. Amount	307	135	209	154	240	93	106	94	212	78
Type2	#11	#12	#13	#14	#15	#16	#17	#18	#19	#20
T.P. Amount	99	133	58	111	69	38	38	103	21	58

Appendix B

Euclidean and Cosine Distance

We denote $Q = (q_1, q_2, \ldots, q_n)$ and $I = (i_1, i_2, \ldots, i_n)$ as the feature representation of query and target images respectively while n is the length of the corresponding feature vectors. Denoting $\|Q\|$ and $\|I\|$ as the norm of Q and I respectively, the Cosine distance between query and target image is defined as

$$Cos(Q, I) = 1 - \frac{\sum_{j=1}^{n}(q_j * i_j)}{\|Q\| * \|I\|} \tag{B.1}$$

Hence, the Euclidean distance between Q and I is computed as follows.

$$\begin{aligned}
Eu(Q, I)^2 &= \textstyle\sum_{j=1}^{n}(q_j - i_j)^2 \\
&= \textstyle\sum_{j=1}^{n}(q_j^2 + i_j^2 - 2(q_j \times i_j)) \\
&= \textstyle\sum_{j=1}^{n} q_j^2 + \sum_{j=1}^{n} i_j^2 - 2\sum_{j=1}^{n}(q_j * i_j) \\
&= \|Q\|^2 + \|I\|^2 - 2\textstyle\sum_{j=1}^{n}(q_j * i_j) \\
&= \|Q\| * \|I\|(\tfrac{\|Q\|}{\|I\|} + \tfrac{\|I\|}{\|Q\|} - 2\tfrac{\sum_{j=1}^{n}(q_j * i_j)}{\|Q\| * \|I\|}) \qquad (B.2) \\
&\propto (\tfrac{\|Q\|}{\|I\|} + \tfrac{\|I\|}{\|Q\|} - 2\tfrac{\sum_{j=1}^{n}(q_j * i_j)}{\|Q\| * \|I\|}) \\
&= (\tfrac{\|Q\|}{\|I\|} + \tfrac{\|I\|}{\|Q\|} - 2 + 2(1 - \tfrac{\sum_{j=1}^{n}(q_j * i_j)}{\|Q\| * \|I\|})) \\
&= \tfrac{\|Q\|}{\|I\|} + \tfrac{\|I\|}{\|Q\|} - 2 + 2Cos(Q, I)
\end{aligned}$$

Hence, when $\|Q\| = \|I\|$, then $\frac{\|Q\|}{\|I\|} + \frac{\|I\|}{\|Q\|} - 2 = 0$ and thus $Eu(Q, I)^2 \propto 2Cos(Q, I)$.In this case, Euclidean distance serves equivalent as the Cosine distance does while only the ranking order of the distances matters for the retrieval system rather than the absolute value of the corresponding distances.

In the following parts, we will further study the relation between Euclidean and Cosine distance when $\|Q\| \neq \|I\|$. Denoting $\alpha = \frac{\|Q\|}{\|I\|}$ and $\beta = \alpha + \frac{1}{\alpha} - 2$, β will be in the case $\alpha > 0$ and $\alpha \neq 1$. Hence, Equation B.2 can be further simplified as,

$$\begin{aligned} Eu(Q,I)^2 &\propto \alpha + \tfrac{1}{\alpha} - 2 + 2Cos(Q,I) \\ &= \beta(1 + \tfrac{2}{\beta}Cos(Q,I)) \\ &\propto 1 + \tfrac{2}{\beta}Cos(Q,I) \end{aligned} \tag{B.3}$$

As shown in Equation B.3, for retrieval purpose where the only the ranking matters, Euclidean distance is correlated with Cosine distance with a factor $\frac{2}{\beta}$. Figure B.1 illustrates the how the coefficient between Euclidean and Cosine distance. As show in the figure, when $\alpha \simeq 1$,in another words when $\|Q\| \simeq \|I\|$, the coefficient of Euclidean and Cosine distance become to nearly positive infinite and thus serves almost equivalent for the retrieval purpose. The shape of the curve plotted in Figure B.1 could be further proved by computing the differential coefficient of $\beta = \alpha + \frac{1}{\alpha} - 2$.

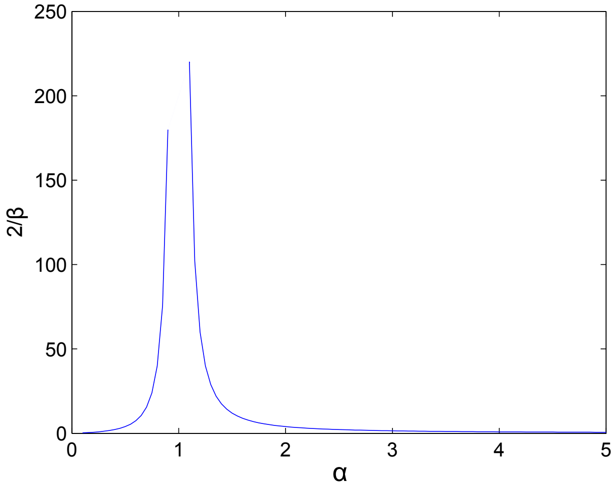


Figure B.1: MAP improvement by Relevance Feedback.

Appendix C

Publications

Refereed journals

- Hongxing Gao, Marçal Rusiñol, Dimosthenis Karatzas and Josep Lladós. Efficient Structure Spotting for Document Images through Spatial Database. *IEEE Transactions on Pattern Recognition*, in reviewing.

Refereed major conferences

- Hongxing Gao, Marçal Rusiñol, Dimosthenis Karatzas and Josep Lladós. Embedding Document Structure to Bag-of-Words through Pair-wise Stable Key-regions. *International Conference on Pattern Recognition*, 2014.
- Hongxing Gao, Marçal Rusiñol, Dimosthenis Karatzas and Josep Lladós. Fast Structural Matching for Document Image Retrieval through Spatial Database. *Document Recognition and Retrieval XXI, Part of the IS&SPIE 26th Annual Symposium on Electronic Imaging*, 2014
- Hongxing Gao, Marçal Rusiñol, Dimosthenis Karatzas, Josep Lladós, Tomokazu Sato, Masakazu Iwamura and Kiochi Kise, Key-region Detection for Document Images Application to Administrative Document Retrieval. *International Conference on Document Analysis and Recognition*, 2013.
- Hongxing Gao, Marçal Rusiñol, Dimosthenis Karatzas, Apostolos Antonacopoulos and Josep Lladós . An Interactive Appearance-based Document Retrieval System for Historical Newspapers , *In the International Conference on Computer Vision Theory and Application*, 2013

Bibliography

[1] K. Taghva, J. Borsack, and A. Condit, "Results of applying probabilistic ir to ocr text," in *The 17th Annual International ACM SIGIR conference on Research and Development in Information Retrieval.* Springer-Verlag New York, Inc., 1994, pp. 202–211.

[2] X. Tong and D. A. Evans, "A statistical approach to automatic ocr error correction in context," in *The 4th workshop on Very Large Corpora*, 1996, pp. 88–100.

[3] R. Jin, A. G. Hauptmann, and C. Zhai, "A content-based probabilistic correction model for ocr document retrieval," in *The 25th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, 2002.

[4] K. Collins-Thompson, C. Schweizer, and S. Dumais, "Improved string matching under noisy channel conditions," in *The 10th International Conference on Information and Knowledge Management.* ACM, 2001, pp. 357–364.

[5] G. Navarro, "Improved approximate pattern matching on hypertext," *Theoretical Computer Science*, vol. 237, no. 1, pp. 455–463, 2000.

[6] ——, "A guided tour to approximate string matching," *ACM Computing Surveys*, vol. 33, no. 1, pp. 31–88, 2001.

[7] D. S. Doermann, E. Rivlin, and I. Weiss, "Logo recognition using geometric invariants," in *The second International Conference on Document Analysis and Recognition.* IEEE, 1993, pp. 894–897.

[8] M. Rusinol and J. Llados, "Logo spotting by a bag-of-words approach for document categorization," in *The 10th International Conference on Document Analysis and Recognition.* IEEE, 2009, pp. 111–115.

[9] M. Rusinol, V. P. DAndecy, D. Karatzas, and J. Lladós, "Classification of administrative document images by logo identification," in *Graphics Recognition: New Trends and Challenges.* Springer, 2013, pp. 49–58.

[10] S. Escalera, A. Fornés, O. Pujol, A. Escudero, and P. Radeva, "Circular blurred shape model for symbol spotting in documents," in *The 16th IEEE International Conference on Image Processing.* IEEE, 2009, pp. 2005–2008.

[11] V. P. Le, M. Visani, C. De Tran, and J. Ogier, "Logo spotting for document categorization," in *The 21st International Conference on Pattern Recognition.* IEEE, 2012, pp. 3484–3487.

[12] M. Ester, H.-P. Kriegel, J. Sander, and X. Xu, "A density-based algorithm for discovering clusters in large spatial databases with noise," in *Knowledge Discovery in Databases*, vol. 96, 1996, pp. 226–231.

[13] V. P. Le, M. Visani, C. D. Tran, and J.-M. Ogier, "Improving logo spotting and matching for document categorization by a post-filter based on homography," in *The 12th International Conference on Document Analysis and Recognition.* IEEE, 2013, pp. 270–274.

[14] R. Jain and D. S. Doermann, "Logo retrieval in document images," in *The 10th International Workshop on Document Analysis Systems*, 2012, pp. 135–139.

[15] J. Fu, J. Wang, and H. Lu, "Effective logo retrieval with adaptive local feature selection," in *The 18th ACM International Conference on Multimedia.* ACM, 2010, pp. 971–974.

[16] S. Belongie, J. Malik, and J. Puzicha, "Shape matching and object recognition using shape contexts," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 24, no. 4, pp. 509–522, 2002.

[17] W. H. Leung and T. Chen, "Retrieval of sketches based on spatial relation between strokes," in *The 10th International Conference on Image Processing*, vol. 1. IEEE, 2002, pp. I–908.

[18] S. Ahmed, M. I. Malik, M. Liwicki, and A. Dengel, "Signature segmentation from document images," in *The 13th International Conference on Frontiers in Handwriting Recognition.* IEEE Computer Society, 2012, pp. 425–429.

[19] S. N. Srihari, S. Shetty, S. Chen, H. Srinivasan, C. Huang, G. Agam, and O. Frieder, "Document image retrieval using signatures as queries," in *The 2nd International Conference on Document Image Analysis for Libraries.* IEEE, 2006, pp. 6–11.

[20] J. T. Favata and G. Srikantan, "A multiple feature/resolution approach to handprinted digit and character recognition," *International Journal of Imaging Systems and Technology*, vol. 7, no. 4, pp. 304–311, 1996.

[21] A. Chalechale, G. Naghdy, and A. Mertins, "Signature-based document retrieval," in *The 3rd IEEE International Symposium on Signal Processing and Information Technology.* IEEE, 2003, pp. 597–600.

[22] G. Zhu, Y. Zheng, D. Doermann, and S. Jaeger, “Signature detection and matching for document image retrieval,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 31, no. 11, pp. 2015–2031, 2009.

[23] J. Hu, R. Kashi, and G. Wilfong, “Document image layout comparison and classification,” in *The 5th International Conference on Document Analysis and Recognition*, 1999, pp. 285–288.

[24] P. Chauvet, J. Lopez-Krahe, E. Taflin, and H. Maître, “System for an intelligent office document analysis, recognition and description,” *Signal Processing*, vol. 32, no. 1, pp. 161–190, 1993.

[25] N. Priyadharshini and M. Vijaya, “Genetic programming for document segmentation and region classification using discipulus,” *International Journal of Advanced Research in Artificial Intelligence*, vol. 2, no. 2, pp. 15–22, 2013.

[26] F. M. Wahl, K. Y. Wong, and R. G. Casey, “Block segmentation and text extraction in mixed text/image documents,” *Computer graphics and image processing*, vol. 20, no. 4, pp. 375–390, 1982.

[27] M. Krishnamoorthy, G. Nagy, S. Seth, and M. Viswanathan, “Syntactic segmentation and labeling of digitized pages from technical journals,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 15, no. 7, pp. 737–747, 1993.

[28] C. L. Tan, B. Yuan, W. Huang, Q. Wang, and Z. Zhang, “Text/graphics separation using agent-based pyramid operations,” in *The 5th International Conference on Document Analysis and Recognition.* IEEE, 1999, pp. 169–172.

[29] S. S. Bukhari, A. Azawi, M. I. Ali, F. Shafait, and T. M. Breuel, “Document image segmentation using discriminative learning over connected components,” in *The 9th International Workshop on Document Analysis Systems.* ACM, 2010, pp. 183–190.

[30] F. Lebourgeois, Z. Bublinski, and H. Emptoz, “A fast and efficient method for extracting text paragraphs and graphics from unconstrained documents,” in *The 11th International Conference on Pattern Recognition.* IEEE, 1992, pp. 272–276.

[31] T. Saitoh and T. Pavlidis, “Page segmentation without rectangle assumption,” in *The 11th International Conference on Pattern Recognition.* IEEE, 1992, pp. 277–280.

[32] M. Lin, J.-R. Tapamo, and B. Ndovie, “A texture-based method for document segmentation and classification,” *South African Computer Journal*, no. 36, pp. p–49, 2006.

[33] R. M. Haralick, “Statistical and structural approaches to texture,” *Proceedings of the IEEE*, vol. 67, no. 5, pp. 786–804, 1979.

[34] S. Lloyd, “Least squares quantization in pcm,” *Information Theory, IEEE Transactions on*, vol. 28, no. 2, pp. 129–137, 1982.

[35] D. W. Ruck, S. K. Rogers, M. Kabrisky, M. E. Oxley, and B. W. Suter, “The multilayer perceptron as an approximation to a bayes optimal discriminant function,” *IEEE Transactions on Neural Networks*, vol. 1, no. 4, pp. 296–298, 1990.

[36] D. S. Bloomberg, “Multiresolution morphological approach to document image analysis,” in *The 1st International Conference on Document Analysis and Recognition*, 1991.

[37] S. S. Bukhari, F. Shafait, and T. M. Breuel, “Improved document image segmentation algorithm using multiresolution morphology,” in *Document Recognition and Retrieval XVIII, Part of the IS&T/SPIE 23th Annual Symposium on Electronic Imaging.* International Society for Optics and Photonics, 2011, pp. 78 740D–78 740D.

[38] J. Ha, R. M. Haralick, and I. T. Phillips, “Recursive xy cut using bounding boxes of connected components,” in *The 3rd International Conference on Document Analysis and Recognition*, vol. 2. IEEE, 1995, pp. 952–955.

[39] F. Cesarini, S. Marinai, and G. Soda, “Retrieval by layout similarity of documents represented with mxy trees,” in *The 5th International Workshop on Document Analysis Systems.* Springer, 2002, pp. 353–364.

[40] P. Duygulu and V. Atalay, “A hierarchical representation of form documents for identification and retrieval,” *International Journal on Document Analysis and Recognition*, vol. 5, no. 1, pp. 17–27, 2002.

[41] S. Baldi, S. Marinai, and G. Soda, “Using tree-grammars for training set expansion in page classification,” in *The 12th International Conference on Document Analysis and Recognition*, vol. 2. IEEE Computer Society, 2003, pp. 829–829.

[42] S. Marinai, E. Marino, and G. Soda, “Layout based document image retrieval by means of xy tree reduction,” in *The 8th International Conference on Document Analysis and Recognition.* IEEE, 2005, pp. 432–436.

[43] J. Liang and D. Doermann, “Logical labeling of document images using layout graph matching with adaptive learning,” in *The 5th International Workshop on Document Analysis Systems.* Springer, 2002, pp. 224–235.

[44] J. Liang, I. T. Phillips, and R. M. Haralick, “Performance evaluation of document structure extraction algorithms,” *Computer Vision and Image Understanding*, vol. 84, no. 1, pp. 144–159, 2001.

[45] J. Van Beusekom, D. Keysers, F. Shafait, and T. M. Breuel, “Distance measures for layout-based document image retrieval,” in *The International Conference on Document Image Analysis for Libraries.* IEEE, 2006, pp. 11–pp.

[46] J. Canny, “A computational approach to edge detection,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, no. 6, pp. 679–698, 1986.

[47] R. Deriche, "Using canny's criteria to derive a recursively implemented optimal edge detector," *International Journal of Computer Vision*, vol. 1, no. 2, pp. 167–187, 1987.

[48] C. Harris and M. Stephens, "A combined corner and edge detector," in *The 4th Alvey vision conference*, vol. 15. Manchester, UK, 1988, p. 50.

[49] D. G. Lowe, "Object recognition from local scale-invariant features," in *The 7th International Conference on Computer Vision*, 1999, pp. 1150–1157.

[50] H. Bay, T. Tuytelaars, and L. Van Gool, "Surf: Speeded up robust features," in *The 9th European Conference on Computer Vision*. Springer, 2006, pp. 404–417.

[51] J. Matas, O. Chum, M. Urban, and T. Pajdla, "Robust wide-baseline stereo from maximally stable extremal regions," *Image and Vision Computing*, vol. 22, no. 10, pp. 761–767, 2004.

[52] D. Zhang and G. Lu, "Review of shape representation and description techniques," *Pattern Recognition*, vol. 37, no. 1, pp. 1–19, 2004.

[53] G.-H. Chuang and C.-C. Kuo, "Wavelet descriptor of planar curves: Theory and applications," *IEEE Transactions on Image Processing*, vol. 5, no. 1, pp. 56–70, 1996.

[54] H. Asada and M. Brady, "The curvature primal sketch," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, no. 1, pp. 2–14, 1986.

[55] K. E. A. van de Sande, T. Gevers, and C. G. M. Snoek, "Evaluating color descriptors for object and scene recognition," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 32, no. 9, pp. 1582–1596, 2010.

[56] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," in *The IEEE Conference on Computer Vision and Pattern Recognition*, vol. 1. IEEE, 2005, pp. 886–893.

[57] M. Brown and D. G. Lowe, "Automatic panoramic image stitching using invariant features," *International Journal of Computer Vision*, vol. 74, no. 1, pp. 59–73, 2007.

[58] Z. Min, Z. Jiguo, and X. Xusheng, "Panorama stitching based on sift algorithm and levenberg-marquardt optimization," *Physics Procedia*, vol. 33, pp. 811–818, 2012.

[59] L. Juan and O. Gwun, "Surf applied in panorama image stitching," in *The 2nd International Conference on Image Processing Theory Tools and Applications*. IEEE, 2010, pp. 495–499.

[60] B. Sirmacek and C. Unsalan, "Urban-area and building detection using sift keypoints and graph theory," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 47, no. 4, pp. 1156–1167, 2009.

[61] D. G. Lowe, “Object recognition from local scale-invariant features,” in *The 7th IEEE International Conference on Computer Vision*, vol. 2. Ieee, 1999, pp. 1150–1157.

[62] M. Lopez-de-la Calleja, T. Nagai, M. Attamimi, M. Nakano-Miyatake, and H. Perez-Meana, “Object detection using surf and superpixels,” *Journal of Software Engineering and Applications*, vol. 6, no. 09, p. 511, 2013.

[63] G. T. Flitton, T. P. Breckon, and N. M. Bouallagu, “Object recognition using 3d sift in complex ct volumes.” in *British Machine Vision Conference*, 2010, pp. 1–12.

[64] G. Flitton, T. P. Breckon, and N. Megherbi, “A comparison of 3d interest point descriptors with application to airport baggage object detection in complex ct imagery,” *Pattern Recognition*, vol. 46, no. 9, pp. 2420–2436, 2013.

[65] J. Knopp, M. Prasad, G. Willems, R. Timofte, and L. Van Gool, “Hough transform and 3d surf for robust three dimensional classification,” in *The 11th European Conference on Computer Vision*. Springer, 2010, pp. 589–602.

[66] B. Li, X. Kong, Z. Wang, and H. Fu, “Sift-based image retrieval combining the distance measure of global image and sub-image,” in *The 5th International Conference on Intelligent Information Hiding and Multimedia Signal Processing*. IEEE, 2009, pp. 706–709.

[67] S. Huang, C. Cai, F. Zhao, D. He, and Y. Zhang, “An efficient wood image retrieval using surf descriptor,” in *International Conference on Test and Measurement*, vol. 2. IEEE, 2009, pp. 55–58.

[68] S. Vitaladevuni, F. Choi, R. Prasad, and P. Natarajan, “Detecting near-duplicate document images using interest point matching,” in *The 21st International Conference on Pattern Recognition (ICPR)*, 2012, pp. 347–350.

[69] D. Smith and R. Harvey, “Document retrieval using sift image features,” *Journal of Universal Computer Science*, vol. 17, no. 1, pp. 3–15, 2011.

[70] O. Augereau, N. Journet, J.-P. Domenger *et al.*, “Semi-structured document image matching and recognition.” in *Document Recognition and Retrieval XX, Part of the IS&T/SPIE 25th Annual Symposium on Electronic Imaging*, 2013.

[71] K. Terasawa and Y. Tanaka, “Slit style hog feature for document image word spotting,” in *The 10th International Conference on Document Analysis and Recognition*, 2009, pp. 116–120.

[72] H. Gao, M. Rusiñol, D. Karatzas, , J. Lladós, T. Sato, M. Iwamura, and K. Kise, “Key-region detection for document images – application to administrative document retrieval,” in *The 12th International Conference on Document Analysis and Recognition*, 2013, pp. 230–234.

[73] J. Li, Z.-G. Fan, Y. Wu, and N. Le, "Document image retrieval with local feature sequences," in *The 10th International Conference on Document Analysis and Recognition*, 2009, pp. 346–350.

[74] G. Meng, N. Zheng, Y. Song, and Y. Zhang, "Document images retrieval based on multiple features combination," in *The 9th International Conference on Document Analysis and Recognition*, vol. 1, 2007, pp. 143–147.

[75] J. Almazán, A. Gordo, A. Fornés, and E. Valveny, "Efficient exemplar word spotting," in *The 23rd British Machine Vision Conference*, 2012, pp. 67.1–67.11.

[76] F. Li and P. Perona, "A bayesian hierarchical model for learning natural scene categories," in *International Conference on Computer Vision and Pattern Recognition*, 2005, pp. 524–531.

[77] R. Shekhar, "Document image retrieval using bag of visual words model," Ph.D. dissertation, International Institute of Information Technology Hyderabad, 2013.

[78] R. Vieux, J. Benois-Pineau, and J.-P. Domenger, *Content based image retrieval using bag-of-regions.* Springer, 2012.

[79] P. F. Felzenszwalb and D. P. Huttenlocher, "Efficient graph-based image segmentation," *International Journal of Computer Vision*, vol. 59, no. 2, pp. 167–181, 2004.

[80] A. Levinshtein, A. Stere, K. N. Kutulakos, D. J. Fleet, S. J. Dickinson, and K. Siddiqi, "Turbopixels: Fast superpixels using geometric flows," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 31, no. 12, pp. 2290–2297, 2009.

[81] S. Sural, G. Qian, and S. Pramanik, "Segmentation and histogram generation using the hsv color space for image retrieval," in *The 10th International Conference on Image Processing*, vol. 2. IEEE, 2002, pp. II–589.

[82] T. Ojala, M. Pietikainen, and T. Maenpaa, "Multiresolution gray-scale and rotation invariant texture classification with local binary patterns," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 24, no. 7, pp. 971–987, 2002.

[83] H. Gao, M. Rusiñol, D. Karatzas, A. Antonacopoulos, and J. Lladós, "An interactive appearance-based document retrieval system for historical newspapers," in *The 8th International Conference on Computer Vision Theory and Applications*, 2013, pp. 84–87.

[84] A. Gordo, F. Perronnin, and E. Valveny, "Large-scale document image retrieval and classification with runlength histograms and binary embeddings," *Pattern Recognition*, vol. 46, no. 7, pp. 1898–1905, 2013.

[85] S. Chen, Y. He, J. Sun, and S. Naoi, "Structured document classification by matching local salient features," in *The 21st International Conference on Pattern Recognition.* IEEE, 2012, pp. 653–656.

[86] J. Kumar, P. Ye, and D. Doermann, "Learning document structure for retrieval and classification," in *The 21st International Conference on Pattern Recognition.* IEEE, 2012, pp. 1558–1561.

[87] ——, "Structural similarity for document image classification and retrieval," *Pattern Recognition Letters*, vol. 43, pp. 119–126, 2014.

[88] Kristo and C. S. Chua, "Optimized window arrangement for spatial pyramid matching," in *The 21st IEEE International Conference on Pattern Recognition.* IEEE, 2014, pp. 1395–1400.

[89] H. Gao, M. Rusiñol, D. Karatzas, and J. Lladós, "Embedding document structure to bag-of-words through pair-wise stable key-regions," in *The 21st IEEE International Conference on Pattern Recognition.* IEEE, 2014, pp. 2903–2908.

[90] M. Muja and D. G. Lowe, "Fast approximate nearest neighbors with automatic algorithm configuration," in *The 6th International Conference on Computer Vision Theory and Application.* INSTICC Press, 2009, pp. 331–340.

[91] A. Kumar, C. V. Jawahar, and R. Manmatha, "Efficient search in document image collections," in *The 8th Asian Conference on Computer Vision, year = 2007, pages = 586–595,.*

[92] D. Zhang, M. Islam, G. Lu, and J. Hou, "Semantic image retrieval using region based inverted file," in *Digital Image Computing: Techniques and Applications*, 2009, pp. 242–249.

[93] H. Jegou, M. Douze, and C. Schmid, "Product quantization for nearest neighbor search," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 33, no. 1, pp. 117–128, 2011.

[94] T. Sato, M. Iwamura, and K. Kise, "Fast and memory efficient approximate nearest neighbor search with distance estimation based on space indexing," IEICE Technical Report, Tech. Rep., Feb. 2013.

[95] J. L. Bentley, "Multidimensional binary search trees used for associative searching," *Communications of the ACM*, vol. 18, no. 9, pp. 509–517, 1975.

[96] R. H. Güting, "An introduction to spatial database systems," *The Very Large Data Bases Journal*, vol. 3, no. 4, pp. 357–399, 1994.

[97] P. Héroux, S. Diana, A. Ribert, and E. Trupin, "Classification method study for automatic form class identification," *The 14th International Conference on Pattern Recognition*, vol. 1, pp. 926–928, 1998.

[98] A. Vadivel, A. Majumdar, and S. Sural, "Performance comparison of distance metrics in content-based image retrieval application," in *International Conference on Information Technology*, 2003, pp. 159–164.

[99] M. Kokare, B. Chatterji, and P. Biswas, “Comparision of similarity metrics for texture image retrieval,” in *Conference on Convergent Technologies for Asia-Pacific Region*, 2003, pp. 571–575.

[100] J. Rocchio, “Relevance feedback in information retrieval,” in *SMART Retrieval System: Experiments in Automatic Document Processing*, 1971, pp. 313–323.

[101] G. Giacinto, “A nearest-neighbor approach to relevance feedback in content based image retrieval,” in *The 6th ACM International Conference on Image and Video retrieval*, 2007, pp. 456–463.

[102] A. Gordo and F. Perronnin, “Asymmetric distances for binary embeddings,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2011, pp. 729–736.

[103] K. Mikolajczyk and C. Schmid, “An affine invariant interest point detector,” in *The 7th European Conference on Computer Vision*, 2002, pp. 128–142.

[104] D. G. Lowe, “Distinctive image features from scale-invariant keypoints,” *International Journal of Computer Vision*, vol. 60, no. 2, pp. 91–110, 2004.

[105] T. Lindeberg, “Feature detection with automatic scale selection,” *International Journal of Computer Vision*, vol. 30, pp. 79–116, 1998.

[106] K. Mikolajczyk and C. Schmid, “A performance evaluation of local descriptors,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 27, no. 10, pp. 1615–1630, 2005.

[107] M. Rusiñol, D. Karatzas, A. Bagdanov, and J. Lladós, “Multipage document retrieval by textual and visual representations,” in *The 21st International Conference on Pattern Recognition*, 2012, pp. 521–524.

[108] G. David, “Distinctive image features from scale-invariant keypoints,” in *International Journal of Computer Vision*, 2004, pp. 91–110.

[109] B. Fulkerson, A. Vedaldi, and S. Soatto, “Localizing objects with smart dictionaries,” in *The 10th European Conference on Computer Vision*, ser. Lecture Notes on Computer Science, 2008, vol. 5302, pp. 179–192.

[110] H. Jégou, M. Douze, and C. Schmid, “Product quantization for nearest neighbor search,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 33, no. 1, pp. 117–128, January 2011.

[111] T. Nakai, K. Kise, and M. Iwamura, “Camera-based document image retrieval as voting for partial signatures of projective invariants,” in *The 8th IEEE International Conference on Document Analysis and Recgonition*, 2005, pp. 379–383.

[112] K. Takeda, K. Kise, and M. Iwamura, “Real-time document image retrieval for a 10 million pages database with a memory efficient and stability improved LLAH,” in *The 11th International Conference on Document Analysis and Recognition*, 2011, pp. 1054–1058.

[113] W. Sun and K. Kise, "Similar manga retrieval using visual vocabulary based on regions of interest," in *The 11th International Conference on Document Analysis and Recgonition*, 2011, pp. 1075–1079.

[114] F. Porikli and T. Kocak, "Fast distance transform computation using dual scan line propagation," in *Real-Time Image Processing*, 2007.

[115] L. Ikonen and P. Toivanen, "Distance and nearest neighbor transforms on gray-level surfaces," *Pattern Recognition Letters*, vol. 28, no. 5, pp. 604 – 612, 2007.

[116] A. Babenko and V. Lempitsky, "The inverted multi-index," in *The 2012 IEEE Conference on Computer Vision and Pattern Recognition*, 2012, pp. 3069–3076.

[117] F. Sebsatiani, "Machine learning in automated text categorization," *Journal ACM Computing Surveys*, vol. 34, no. 1, pp. 1–47, March 2002.

[118] A. Bagdanov, "Fine-grained document genre classification using first order random graphs," in *The 6th International Conference on Document Analysis and Recognition*, 2001, pp. 79–83.

[119] C. Shin and D. S. Doermann, "Document image retrieval based on layout structural similarity." in *International Conference on Image Processing, Computer Vision, & Pattern Recognition.* Citeseer, 2006, pp. 606–612.

[120] J. Zobel and A. Moffat, "Inverted files for text search engines," *ACM Computing Surveys*, vol. 38, no. 2, 2006.

[121] M. Aly, M. Munich, and P. Perona, "Indexing in large scale image collections: Scaling properties and benchmark," in *2011 IEEE Workshop on Applications of Computer Vision*, 2011, pp. 418–425.

[122] G. Salton and C. Buckley, "Term-weighting approaches in automatic text retrieval," in *Information Processing and Management*, 1988, pp. 513–523.

[123] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman, "The PASCAL Visual Object Classes Challenge 2011 (VOC2011) Results," http://www.pascal-network.org/challenges/VOC/voc2011/workshop/index.html.

[124] Y. Manolopoulos, A. Nanopoulos, A. N. Papadopoulos, and Y. Theodoridis, *R-Trees: Theory and Applications.* Secaucus, NJ, USA: Springer-Verlag New York, Inc., 2005.

[125] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman, "The pascal visual object classes (voc) challenge," *International Journal of Computer Vision*, vol. 88, no. 2, pp. 303–338, 2010.

[126] T. Matsuda, M. Iwamura, and K. Kise, "Performance improvement in local feature based camera-captured character recognition," in *The 11th IAPR International Workshop on Document Analysis Systems.* IEEE, 2014, pp. 196–201.