SUPERVISED AND UNSUPERVISED SEGMENTATION OF TEXTURED IMAGES BY EFFICIENT MULTI-LEVEL PATTERN CLASSIFICATION

Jaime Christian Meléndez Rodríguez

ISBN: 978-84-693-7671-3
Dipòsit Legal: T-1750-2010

UNIVERSITAT ROVIRA I VIRGILI
SUPERVISED AND UNSUPERVISED SEGMENTATION OF TEXTURED IMAGES BY EFFICIENT MULTI-LEVEL PATTERN CLASSIFICATION
Jaime Christian Meléndez Rodríguez
ISBN:978-84-693-7671-3/DL:T-1750-2010

Jaime Christian Meléndez Rodríguez

SUPERVISED AND UNSUPERVISED SEGMENTATION OF TEXTURED IMAGES BY EFFICIENT MULTI-LEVEL PATTERN CLASSIFICATION

Ph.D. Thesis

Supervisors:
Dr. Domènec Puig Valls
Dr. Miguel Ángel García García

Department of Computer Engineering and Mathematics

UVERSITAT ROVIRA I VIRGILI

Tarragona
2010

UNIVERSITAT ROVIRA I VIRGILI
SUPERVISED AND UNSUPERVISED SEGMENTATION OF TEXTURED IMAGES BY EFFICIENT MULTI-LEVEL PATTERN CLASSIFICATION
Jaime Christian Meléndez Rodríguez
ISBN:978-84-693-7671-3/DL:T-1750-2010

UNIVERSITAT ROVIRA I VIRGILI
SUPERVISED AND UNSUPERVISED SEGMENTATION OF TEXTURED IMAGES BY EFFICIENT MULTI-LEVEL PATTERN CLASSIFICATION
Jaime Christian Meléndez Rodríguez
ISBN:978-84-693-7671-3/DL:T-1750-2010

Supervisors' authorization

(not available in this electronic version)

UNIVERSITAT ROVIRA I VIRGILI
SUPERVISED AND UNSUPERVISED SEGMENTATION OF TEXTURED IMAGES BY EFFICIENT MULTI-LEVEL PATTERN CLASSIFICATION
Jaime Christian Meléndez Rodríguez
ISBN:978-84-693-7671-3/DL:T-1750-2010

UNIVERSITAT ROVIRA I VIRGILI
SUPERVISED AND UNSUPERVISED SEGMENTATION OF TEXTURED IMAGES BY EFFICIENT MULTI-LEVEL PATTERN CLASSIFICATION
Jaime Christian Meléndez Rodríguez
ISBN:978-84-693-7671-3/DL:T-1750-2010

To the one who owes me everything.

UNIVERSITAT ROVIRA I VIRGILI
SUPERVISED AND UNSUPERVISED SEGMENTATION OF TEXTURED IMAGES BY EFFICIENT MULTI-LEVEL PATTERN CLASSIFICATION
Jaime Christian Meléndez Rodríguez
ISBN:978-84-693-7671-3/DL:T-1750-2010

UNIVERSITAT ROVIRA I VIRGILI
SUPERVISED AND UNSUPERVISED SEGMENTATION OF TEXTURED IMAGES BY EFFICIENT MULTI-LEVEL PATTERN CLASSIFICATION
Jaime Christian Meléndez Rodríguez
ISBN:978-84-693-7671-3/DL:T-1750-2010

Acknowledgments

The formalism indicates that the first lines of these acknowledgements should be for my advisors... And they will, fortunately, not because of the formalism, but because they deserve them. Indeed, I will always remember some simple, yet very important moments, such as my first day in Tarragona, when Miguel Angel came to pick me up at the train station and then showed me the city, or those days in hospital, with Domènec coming to visit and assist me. So thank you both for more than just academic supervision.

Thanks to Prof. Maria Petrou for her hospitality and for a fruitful research period at Imperial College.

Thanks to the members of the Intelligent Robotics and Computer Vision group for their friendship. Thanks to Rodrigo and Marcela for all their help and for those great moments on faraway lands. Thanks to Ling for her kindness. Thanks to Tomás for assisting this novel, pseudo informatics engineer and for his valuable insight about the Catalan culture.

Thanks to Alfonso and his family for becoming my family while in London.

Thanks to Lourdes and Nelson for being my parachutes in the deepest of my falls.

Thanks to Jessica and Carlo for some big lessons learnt on the day by day.

Thanks to my fair, six-string-ladies, Clara and Susana, for their encouragement and patience in bringing me to a different kind of abstraction and for sharing with me many beautiful signals to process.

Thanks to Jessica, who pulled the trigger for this adventure to begin at the always difficult Spanish embassy in my country.

Thanks to Alfredo and Sabine for their many straightforward, and therefore brilliant, demonstrations of what a friend is.

UNIVERSITAT ROVIRA I VIRGILI
SUPERVISED AND UNSUPERVISED SEGMENTATION OF TEXTURED IMAGES BY EFFICIENT MULTI-LEVEL PATTERN CLASSIFICATION
Jaime Christian Meléndez Rodríguez
ISBN:978-84-693-7671-3/DL:T-1750-2010

Finally, thanks to my people in Trujillo, especially to those who cry, laugh and keep dreaming together with me and because of whom I am still alive.

UNIVERSITAT ROVIRA I VIRGILI
SUPERVISED AND UNSUPERVISED SEGMENTATION OF TEXTURED IMAGES BY EFFICIENT MULTI-LEVEL PATTERN CLASSIFICATION
Jaime Christian Meléndez Rodríguez
ISBN:978-84-693-7671-3/DL:T-1750-2010

Abstract

This thesis proposes new, efficient methodologies for supervised and unsupervised texture segmentation based on supervised pixel-based texture classification. The proposed classification scheme follows a multi-level, top-down approach in order to classify every image pixel by processing texture features obtained after evaluating a set of texture methods over multiple windows of different size.

For the supervised case, two classification methods based on k-nearest neighbors (KNN) and support vector machines (SVMs) are developed. Two alternative approaches relying on clustering algorithms are proposed in order to determine appropriate sets of prototypes for the KNN classifier. The first approach consists of a resolution-driven clustering algorithm, whereas the second approach exploits the optimality of the normalized cut criterion. For the SVM-based classifier, a one-against-one multiclass extension with probability estimates is introduced. In order to configure the developed classifiers with suitable parameters, a parameter selection algorithm based on the classification of the training set is also proposed.

Further enhancements are then achieved by adding three strategies for efficiency improvement. The first strategy utilizes a heuristic measure in order to reduce the number of prototypes evaluated by the KNN classifier. The second strategy yields a single support vector for each of the binary SVMs that make up the multiclass SVM-based classifier. The third strategy carries out feature selection by adapting the heuristic applied for prototype reduction.

For the unsupervised case, the key point of this thesis is the inclusion of supervision by extending the previous methodology developed for the supervised case. As a result, a two-stage unsupervised segmentation technique is proposed. In the first stage, a pattern discovery algorithm determines the texture patterns present in a given image.

UNIVERSITAT ROVIRA I VIRGILI
SUPERVISED AND UNSUPERVISED SEGMENTATION OF TEXTURED IMAGES BY EFFICIENT MULTI-LEVEL PATTERN CLASSIFICATION
Jaime Christian Meléndez Rodríguez
ISBN:978-84-693-7671-3/DL:T-1750-2010

In the second stage, a supervised pixel-based classifier is trained with those patterns and is used to classify every image pixel, thus yielding the final segmentation. In this way, the original unsupervised problem is effectively transformed into a supervised one.

Experiments on a wide variety of textured images show that the proposed methodologies achieve better performance than several state-of-the-art and traditional supervised classifiers and unsupervised segmenters in terms of both segmentation quality and processing time.

Keywords: supervised texture segmentation, unsupervised texture segmentation, supervised pixel-based classification, multi-sized evaluation windows, multi-level classification, prototype-based classification, support vector machines, clustering.

UNIVERSITAT ROVIRA I VIRGILI
SUPERVISED AND UNSUPERVISED SEGMENTATION OF TEXTURED IMAGES BY EFFICIENT MULTI-LEVEL PATTERN CLASSIFICATION
Jaime Christian Meléndez Rodríguez
ISBN:978-84-693-7671-3/DL:T-1750-2010

Contents

UNIVERSITAT ROVIRA I VIRGILI
SUPERVISED AND UNSUPERVISED SEGMENTATION OF TEXTURED IMAGES BY EFFICIENT MULTI-LEVEL PATTERN CLASSIFICATION
Jaime Christian Meléndez Rodríguez
ISBN:978-84-693-7671-3/DL:T-1750-2010

UNIVERSITAT ROVIRA I VIRGILI
SUPERVISED AND UNSUPERVISED SEGMENTATION OF TEXTURED IMAGES BY EFFICIENT MULTI-LEVEL PATTERN CLASSIFICATION
Jaime Christian Meléndez Rodríguez
ISBN:978-84-693-7671-3/DL:T-1750-2010

UNIVERSITAT ROVIRA I VIRGILI
SUPERVISED AND UNSUPERVISED SEGMENTATION OF TEXTURED IMAGES BY EFFICIENT MULTI-LEVEL PATTERN CLASSIFICATION
Jaime Christian Meléndez Rodríguez
ISBN:978-84-693-7671-3/DL:T-1750-2010

UNIVERSITAT ROVIRA I VIRGILI
SUPERVISED AND UNSUPERVISED SEGMENTATION OF TEXTURED IMAGES BY EFFICIENT MULTI-LEVEL PATTERN CLASSIFICATION
Jaime Christian Meléndez Rodríguez
ISBN:978-84-693-7671-3/DL:T-1750-2010

List of Figures

UNIVERSITAT ROVIRA I VIRGILI
SUPERVISED AND UNSUPERVISED SEGMENTATION OF TEXTURED IMAGES BY EFFICIENT MULTI-LEVEL PATTERN CLASSIFICATION
Jaime Christian Meléndez Rodríguez
ISBN:978-84-693-7671-3/DL:T-1750-2010

UNIVERSITAT ROVIRA I VIRGILI
SUPERVISED AND UNSUPERVISED SEGMENTATION OF TEXTURED IMAGES BY EFFICIENT MULTI-LEVEL PATTERN CLASSIFICATION
Jaime Christian Meléndez Rodríguez
ISBN:978-84-693-7671-3/DL:T-1750-2010

ISBN:978-84-693-7671-3/DL:T-1750-2010

List of Tables

UNIVERSITAT ROVIRA I VIRGILI
SUPERVISED AND UNSUPERVISED SEGMENTATION OF TEXTURED IMAGES BY EFFICIENT MULTI-LEVEL PATTERN CLASSIFICATION
Jaime Christian Meléndez Rodríguez
ISBN:978-84-693-7671-3/DL:T-1750-2010

UNIVERSITAT ROVIRA I VIRGILI
SUPERVISED AND UNSUPERVISED SEGMENTATION OF TEXTURED IMAGES BY EFFICIENT MULTI-LEVEL PATTERN CLASSIFICATION
Jaime Christian Meléndez Rodríguez
ISBN:978-84-693-7671-3/DL:T-1750-2010

Chapter 1

Introduction

Computer vision aims at analyzing and interpreting the information present in images, which may come from many sources, such as video sequences, views from multiple cameras or multidimensional data from medical scanners. Since computer vision covers a wide range of topics, it is often related to other disciplines, such as signal processing and artificial intelligence. Examples of application of computer vision include: robot control, visual surveillance, medical image analysis, etc.

One of the main objectives of image understanding is to recognize the contents of a given scene, which can be categorized as a *high-level task*, since it involves identifying objects and even reasoning about them based on their various attributes. In turn, *low-level tasks*, such as image segmentation or classification, must be carried out in order to measure these attributes.

While *image segmentation* consists of splitting a given image into homogeneous regions, *image classification* aims at identifying the class to which each of those regions belongs by considering a set of previously selected classes of interest. Low-level features, such as color or texture, are commonly taken into account for both image segmentation and classification.

Texture is an important *visual cue* necessary for extracting low-level features. Unfortunately, due to its natural complexity, there is neither a unified definition

nor a general representation of texture, as it occurs with other visual cues, such as color. In fact, a wide variety of *texture feature extraction methods* (*texture methods* in short) have been proposed in the literature in order to extract features that allow for the characterization of different texture patterns. A wide variety of techniques for classifying or segmenting a given image based on the outcome of these texture methods have been proposed as well. Their performance usually depends on the texture content and the type of processing they apply.

One of the main drawbacks of the image classification and segmentation techniques based on texture features that have been proposed so far is that they do not provide an efficient characterization of the analyzed texture patterns. Therefore, in order to achieve accurate results, computationally expensive strategies are utilized, which make them unsuitable for tasks where time constraints are important, such as robot exploration. In consequence, it is necessary to develop new classification and segmentation schemes that provide a more efficient methodology in order to achieve better results than with previous techniques, in terms of both accuracy and processing time.

In this dissertation, the main aspects related to texture image classification and segmentation are analyzed, and new techniques are proposed with the final goal of developing an image classifier and segmenter that outperforms the current approaches.

1.1 Problem Statement

As mentioned above, the problem of image classification consists of identifying the patterns present in a given image. In the most general case, *pixel-based classification* may be necessary in order to identify the pattern to which every image pixel belongs. Obviously, classifying every pixel leads to a segmentation of the whole image as a collateral effect. Therefore, pixel-based classification is a possible way to perform image segmentation.

Image classification can be either supervised or unsupervised, depending on whether prior knowledge about the image patterns is available or not. *Supervised image classification* aims at recognizing the patterns present in an input image given a set of known models of interest. Alternatively, the goal of *unsupervised image classification* is to cluster image pixels into salient homogeneous regions according to some features, although it does not pretend to recognize the model associated with each of them.

In general, image classification techniques use one or several visual cues in order to obtain low-level features that allow for the identification of the different patterns present in the image. The most commonly used cues are: shape, color and texture (Sonka et al., 1993).

Shape is one of the main characteristics of objects and has been extensively used in content-based image retrieval systems. Numerous shape analysis and representation techniques have been proposed in the literature. They include: chain codes, polygonal approximations, signatures, edge histograms, Fourier descriptors, moments and image morphology, etc. Some of them require images to have proper segmentation information available. Thus, they are not applied on their own.

Color is probably the most studied visual cue in computer vision. Its use for low-level computer vision tasks, such as segmentation and classification, has become increasingly frequent. Moreover, different well-known color representations, such as RGB, which is oriented to displaying images, or CIELAB, which aims at obtaining estimations of perceptual color differences, have been proposed.

Texture is the third major visual cue. It can be observed in the structural patterns of surfaces of both natural and artificial objects, and may intuitively be interpreted as a measure of surface properties, such as smoothness, coarseness, regularity, etc. Thus, many common visual tasks can benefit from the use of low-level descriptions based on the quantification of the texture content. Unfortunately, texture is the visual cue most difficult to model, being intrinsically noisy by nature and affected by various

UNIVERSITAT ROVIRA I VIRGILI
SUPERVISED AND UNSUPERVISED SEGMENTATION OF TEXTURED IMAGES BY EFFICIENT MULTI-LEVEL PATTERN CLASSIFICATION
Jaime Christian Meléndez Rodríguez
ISBN:978-84-693-7671-3/DL:T-1750-2010

external factors, such as illumination, rotation and scale, which alter its perception.

Furthermore, due to the intrinsic complexity, a formal approach or precise definition of texture is not available. In fact, researchers tend to define texture depending on the particular application for which textural information will be used. For instance, a dictionary definition of texture is "a structure composed of closely interwoven elements." This description is intimately tied to the idea of *texture resolution*, which might be thought of as the average amount of pixels corresponding to each discernable texture element. If this number is large enough, it would be possible to describe the individual elements in some detail. However, as this number approaches unity, it becomes increasingly difficult to characterize them.

Another definition is provided by Pickett (1970), who considers texture as consisting of a large number of elements, each visible to some degree, and densely and evenly (possibly randomly) arranged over the field of view. This definition corresponds to the *statistical approach*. The statistical approach aims at globally characterizing of texture through statistical properties of the spatial distribution of gray levels. The key aspect of this approach is that texture descriptors solely rely on pixel properties, without taking into account the use of elements or subregions.

Alternatively, the *structural approach* (Roselfeld and Lipkin, 1970) suggests that texture is an arrangement of a set of spatial subpatterns according to certain placement rules. Subpatterns, in general, are also constituted by smaller subpatterns positioned according to some placement rules, which yields a recursive approach that aims at capturing the hierarchical structure of scenes. Both the subpatterns and their placement may be statistically characterized as well.

Besides the lack of a unified definition of texture, there is not a taxonomy of different types of textures. A classification closely related to the approaches mentioned above was proposed by Haralick and Shapiro (1992). It groups textures into two major categories according to their internal composition: *microtextures*, which are characterized by qualitative properties observed in small regions, and *macrotextures*,

UNIVERSITAT ROVIRA I VIRGILI
SUPERVISED AND UNSUPERVISED SEGMENTATION OF TEXTURED IMAGES BY EFFICIENT MULTI-LEVEL PATTERN CLASSIFICATION
Jaime Christian Meléndez Rodríguez
ISBN:978-84-693-7671-3/DL:T-1750-2010

which are made up of primitives with specific properties placed according to some rules. In this sense, *statistical analysis* should be suitable for fine microtextures, whereas *structural analysis* should be suitable for coarse macrotextures.

In addition, several attempts to provide a general texture representation model based on psychophysical and psychological factors exist. One of the early psychophysical approaches is due to Julesz (1975). *Julesz's conjecture* states that two textures cannot be instantly discriminated if their second-order statistics are identical. However, since there are specific instances where textures with the same second-order statistics can be discriminated on the basis of local geometrical features (e.g., colinearity, closure), Julesz reformulated his initial hypothesis and developed the *texton* theory (Julesz, 1981, 1986) in order to explain those cases. According to that theory, textons are the units of *preattentive* human texture perception, and correspond to local visual features, such as end-points of line segments, corners and others. Following this line, Vanrell (1996) proposed a model of preattentive texture perception represented by a four-dimensional space whose axes are interpreted in terms of contrast, scale and orientation of the textural elements.

Since preattentive vision does not suffice in cases where image components are not instantaneously discriminable, such as when checking the fidelity of manufacturing processes or inspecting a medical image, texture representation models for *attentive* perception have also been studied. Tamura et al. (1978) proposed to assign a perceptual rating value to textures according to six visual properties, namely: coarseness, contrast, directionality, line-likeness, regularity and roughness. Amadasun and King (1989) conducted similar ranking experiments according to visual properties such as coarseness, contrast, busyness, complexity and texture strength.

Rao and Lohse (1993) defined a three-dimensional model of texture in analogy to those existing for color, where the identified dimensions correspond to repetitiveness versus irregularity, directional versus non-directional, and structurally complex versus simple. Later, this model was refined in (Rao and Lohse, 1996) through new

UNIVERSITAT ROVIRA I VIRGILI
SUPERVISED AND UNSUPERVISED SEGMENTATION OF TEXTURED IMAGES BY EFFICIENT MULTI-LEVEL PATTERN CLASSIFICATION
Jaime Christian Meléndez Rodríguez
ISBN:978-84-693-7671-3/DL:T-1750-2010

psychophysic experiments, yielding the following three-dimensional representation: repetitive versus non-repetitive, high-contrast and non-directional versus low-contrast and directional, and granular, coarse and low-complexity versus non-granular, fine and high-complexity. Heaps and Handel (1999) conducted further studies using the same methodology as Rao and Lohse, but arrived at different conclusions. They also stated that it is not possible to reliably associate a single visual property with each dimension of the texture space.

In (Long and Leow, 2001), a four-dimensional texture space was determined based on information measurements. It was shown that this space has a high degree of consistency with respect to previously defined texture spaces. However, no cues about the meaning of each dimension were provided.

One of the main problems for defining a perceptual texture space is the subjectivity when judging texture similarity according to some perceptual feature, since different people can define that feature in different ways. Furthermore, the similarity between two textures may not only depend on the textures themselves, but also on the context in which that similarity is assessed. Another problem is the difficulty to establish individual per-feature rankings and then try to combine them, since the relative scale between those features is unknown. Finally, some perceptual features are correlated and need to be combined in order to define relevant texture dimensions, which leads to inconsistencies when determining a minimum, representative feature set. Enough insight on how to obtain a computational model capable of integrating features in order to produce relevant variations along specific texture dimensions has not been attained yet.

As a result, a wide variety of alternative techniques referred to as *texture methods* (Section 2.1) have been proposed in the literature in order to obtain computational measures that characterize textures. Texture methods are based on mathematical or physical formulations and are expected to generate different measures when applied to different texture patterns. They can also be defined in order to provide quantitative

measures that represent perceptual features (e.g., Abbadeni et al., 2000; Long and Leow, 2001). Since texture discrimination requires a large enough number of elements to be considered, texture methods are evaluated over the pixels contained in a certain neighborhood (evaluation window). These neighborhoods are typically square and their size is commonly defined on an experimental basis.

Leaving aside the definitions and different ways to characterize textures, four major application domains related to texture analysis exist (Tuceryan and Jain, 1998): texture classification, texture segmentation, texture synthesis and shape-from-texture. Each domain is described below.

Texture classification aims at determining the class (texture) to which a given image or image region (in the case of multi-textured images) belongs. Classification methods can be either *supervised* or *unsupervised*. In supervised classification, examples of each texture class are provided as a training set. A supervised classifier is trained using this set to learn a characterization for each texture class. On the other hand, unsupervised classification does not require prior knowledge and is able to automatically discover different classes from input textures.

The majority of classification methods consist of two stages: *feature extraction*, which yields a representation of each texture class in terms of feature measures, and *classification*, in which classifiers are trained to determine the class of each input texture based on measures of selected texture features. In this case, a classifier is a function that takes the selected features as input and the texture classes as output. The most important supervised classification methods will be reviewed in Section 2.3.

Texture segmentation aims at partitioning a given image into disjoint regions of uniform texture based on texture properties. The results of segmentation can be the input of further image processing and analysis tasks (e.g., classification). Similarly to classification, texture segmentation also involves extracting features and deriving metrics to segregate textures. However, segmentation is generally more complex, since the boundaries that separate different texture regions must be detected in addition

to the texture within each region.

Texture segmentation can be either supervised or unsupervised depending on whether prior knowledge about the image or texture classes is available. Supervised texture segmentation identifies and separates regions that match texture properties shown in the training textures. Unsupervised segmentation must recover the texture classes from an image before separating them into regions. Compared to supervised segmentation, the unsupervised case may be more flexible, although it is generally more computationally expensive. The most representative unsupervised segmentation methods are reviewed in Section 2.4.

Texture synthesis consists of generating textured images from texture descriptions obtained from the given samples. A synthetic texture should differ from the samples, but should have perceptually identical texture characteristics. Compared to texture classification and segmentation, texture synthesis poses a bigger challenge on texture analysis, since reproducing textures is generally more difficult than discriminating them, since it requires more detailed texture descriptions. Some recent work in texture synthesis is reported in (Papari and Petkov, 2009; Sinha and Gupta, 2010; Peyrė, 2010).

Shape-from-texture is the problem of estimating a three-dimensional surface shape by analyzing texture properties of a two-dimensional image. The weak homogeneity or isotropy of a texture is likely to provide a shape cue. For instance, texture gradient, which is usually the result of perspective projection when the surface is viewed from a slant, is used to infer the parameters of surface shape or the underlying perspective transformation. Shape-from-texture has been used for recovering true surface orientation, reconstructing surface shape and inferring the three-dimensional layout of objects in many applications. Some representative work on this topic can be found in (Blostein and Ahuja, 1989; Super and Bovik, 1995; Clerc and Mallat, 2002; Farid and Kosecka, 2007).

1.2 Research Focus

As previously stated, the objective of pixel-based image classification is to identify the pattern associated with every pixel of a given image. In the case of textured images, patterns correspond to texture classes. Therefore, the problem is known as *pixel-based texture classification* (Garcia and Puig, 2003) and leads to *texture segmentation* as a collateral effect. In addition, if information about the texture classes that may be found in the image is utilized in the classification process, the problem is referred to as *supervised pixel-based texture classification*.

This dissertation aims at the development of a new methodology for supervised pixel-based texture classification with the final purpose of improving both supervised and unsupervised texture segmentation in terms of segmentation quality, as well as processing time.

Currently, most texture classifiers and segmenters evaluate texture methods over single square windows of a predefined size in order to obtain texture features that are then processed by pattern recognition techniques. Regarding the effects of both the shape and size of the evaluation windows, Garcia-Sevilla and Petrou (2000) concluded that texture characterization is much more influenced by the window size than by its shape. Later, Puig and Garcia (2003) showed that different evaluation window sizes are beneficial when dealing with different texture methods and classification problems. In practice, the optimal evaluation window size is usually determined empirically (e.g., Jain and Karu, 1996; Kim et al., 2002; Muneeswaran et al., 2006).

The main problem regarding the definition of a proper evaluation window size is that complementary tasks, such as texture classification and texture segmentation, have contradictory requirements to perform well. Thus, while texture classification requires rather large evaluation windows in order to obtain meaningful and reliable descriptions of their content, texture segmentation requires small evaluation windows in order to accurately locate the boundaries between regions of different texture.

However, if those evaluation windows are too small, misclassification in boundary zones may occur, which may lead to improper segmentation. Trying to partially fulfill those requirements, some multi-window approaches have been proposed (e.g., Puig and Garcia, 2003; Melendez et al., 2008; Rao et al., 2009). Following this line, the segmentation quality has been improved in this thesis by evaluating texture methods over multiple windows of different size.

After texture features have been extracted, due to the nature of supervised pixel-based classification, a single feature vector is associated with every pixel of both the input image to be classified and the training images corresponding to the different texture patterns of interest. Regarding the latter, this poses the problem of how to summarize all the available information in order to efficiently model each texture pattern, since every training image potentially generates as many *prototypes* or *representatives* to compare with as pixels. However, the use of so many of those elements may not be necessary in order to achieve satisfactory results and, furthermore, can be prohibitive in real applications in terms of both computational time and memory usage. In this thesis, new methodologies for efficient pattern modeling and classification have been developed.

The general scheme for supervised pixel based classification followed in this thesis is shown in Figure 1.1. It is constituted by two main stages: the first stage, Figure 1.1(a), aims at obtaining texture features from the given input images by evaluating a set of texture methods over multiple windows of different size. A feature selection algorithm may be applied in order to determine the most relevant texture methods. The second stage, Figure 1.1(b), performs the classification task by processing the extracted texture features, taking the previously learned texture models into account.

Regarding unsupervised texture segmentation, an inherent problem of current approaches is the absence of prior knowledge related to the texture patterns present in the images to be segmented, since, by definition, such information is not available.

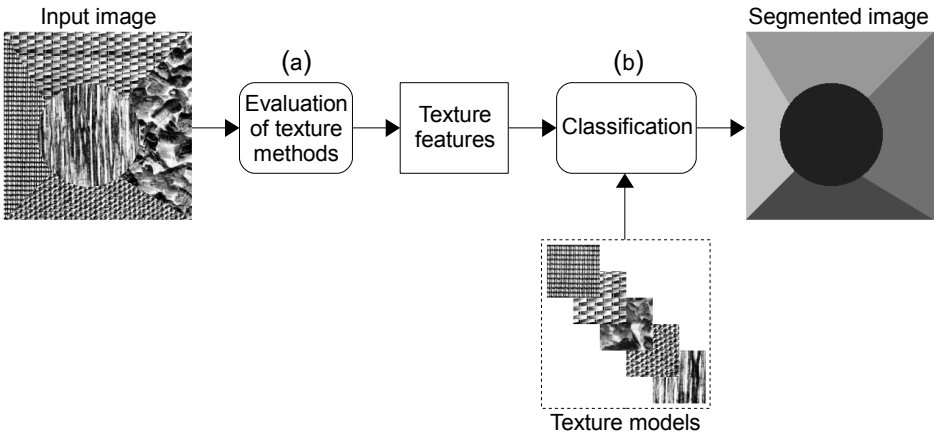

Figure 1.1: Proposed scheme for supervised texture segmentation.

As a consequence, their supervised counterparts, which are specifically trained to identify a given group of patterns, usually perform better in terms of segmentation accuracy as demonstrated in previous works (e.g., Garcia and Puig, 2003). Taking this observation into account, this thesis proposes to introduce supervision into the unsupervised problem by extending the methodology for supervised pixel-based classification described above.

The scheme for the aforementioned extension is shown in Figure 1.2. It consists of three stages: the first stage, Figure 1.2(a), evaluates a set of texture methods in order to obtain low-level texture features from the input images, in the same way as the first scheme (Figure 1.1) does. The second stage, Figure 1.2(b), consists of a *pattern discovery* process that aims at extracting the meaningful texture patterns of the input image, which will then be used to train a supervised classifier. The last stage, Figure 1.2(c), carries out the pixel-based classification task by taking into account the previously obtained features and texture models. This yields the final segmented image.

In summary, this dissertation presents a supervised pixel-based texture classifier that evaluates a set of texture feature extraction methods over multiple windows of

UNIVERSITAT ROVIRA I VIRGILI
SUPERVISED AND UNSUPERVISED SEGMENTATION OF TEXTURED IMAGES BY EFFICIENT MULTI-LEVEL PATTERN CLASSIFICATION
Jaime Christian Meléndez Rodríguez
ISBN:978-84-693-7671-3/DL:T-1750-2010

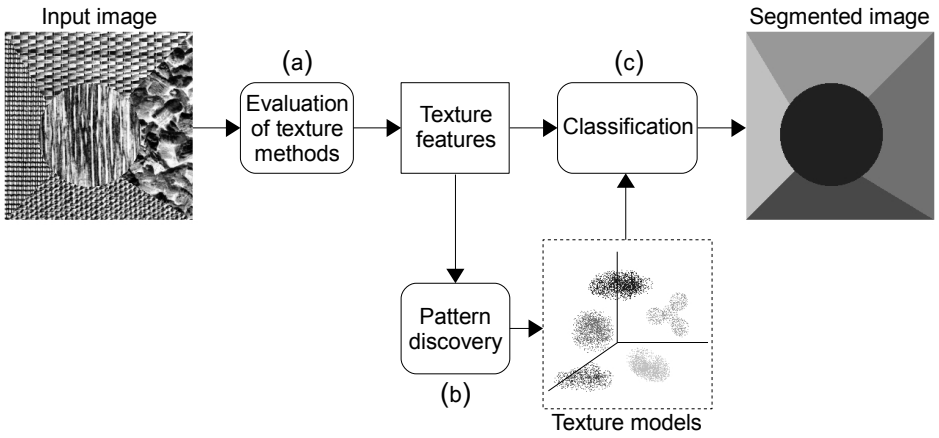

Figure 1.2: Proposed scheme for unsupervised texture segmentation.

different size and relies on efficient pattern learning schemes in order to achieve image segmentation. This new scheme is shown to outperform current supervised texture segmentation approaches both in terms of accuracy and processing time. Furthermore, a new methodology for extending this supervised classifier to the unsupervised domain, thus benefiting from the advantages introduced by supervision, is presented. The resulting unsupervised technique is shown to outperform current unsupervised texture segmentation approaches.

1.3 Organization of the Dissertation

The remainder of this dissertation is organized as follows. Chapter 2 highlights previous related work regarding the topics dealt with in this thesis.

Chapter 3 describes a new supervised texture segmentation technique based on a pixel-based texture classifier that achieves efficient pattern learning by means of a multi-level classification approach that relies on either prototype-based or support vector-based classification techniques and on texture features extracted over multiple windows of different size. A computational process for automatically setting the

classifier's parameters is also proposed. Finally, the processing time of the proposed classifier is further lowered by introducing prototype reduction and feature selection heuristics, as well as by summarizing the support vector information.

Chapter 4 presents a new unsupervised texture segmentation technique that derives from the extension of the supervised classifier outlined in the previous chapter. In order to obtain suitable training patterns for this classifier, a pattern discovery stage that relies on clustering algorithms is applied. Several modifications to some well-known, general purpose clustering algorithms are introduced in order to alleviate their major drawbacks while keeping efficiency. A new measure for evaluating image segmentation quality inspired by the classification rate of pixel-based classifiers is also proposed.

Chapter 5 presents an experimental validation of the techniques described in the previous chapters. Comparisons to well-known supervised and unsupervised texture segmenters, as well as between the developed supervised and unsupervised segmentation techniques, are carried out.

This dissertation concludes with Chapter 6, which summarizes the whole work, highlighting its main contributions and suggesting future lines of research.

UNIVERSITAT ROVIRA I VIRGILI
SUPERVISED AND UNSUPERVISED SEGMENTATION OF TEXTURED IMAGES BY EFFICIENT MULTI-LEVEL PATTERN CLASSIFICATION
Jaime Christian Meléndez Rodríguez
ISBN:978-84-693-7671-3/DL:T-1750-2010

UNIVERSITAT ROVIRA I VIRGILI
SUPERVISED AND UNSUPERVISED SEGMENTATION OF TEXTURED IMAGES BY EFFICIENT MULTI-LEVEL PATTERN CLASSIFICATION
Jaime Christian Meléndez Rodríguez
ISBN:978-84-693-7671-3/DL:T-1750-2010

Chapter 2

Previous Related Work

This chapter reviews selected previous work related to the texture analysis problems studied in this dissertation. It has been divided into four sections concerning the following topics: texture feature extraction methods (Section 2.1), determining the optimal evaluation window for texture segmentation (Section 2.2), supervised texture classification (Section 2.3) and unsupervised texture segmentation (Section 2.4).

2.1 Texture Feature Extraction Methods

As stated in Chapter 1, the lack of a unified texture representation model has motivated the development of a large number of methods to obtain computational measures that characterize textures. These methods can be divided into four major categories: statistical methods, signal processing methods, structural methods and model-based methods. These categories will be surveyed in the following sections and references about their utilization in the considered application domains will be provided. Other interesting surveys can be found in (Reed and du Buf, 1993; Tuceryan and Jain, 1998; Randen and Husøy, 1999; Zhang and Tan, 2002; Petrou and Garcia Sevilla, 2006).

UNIVERSITAT ROVIRA I VIRGILI
SUPERVISED AND UNSUPERVISED SEGMENTATION OF TEXTURED IMAGES BY EFFICIENT MULTI-LEVEL PATTERN CLASSIFICATION
Jaime Christian Meléndez Rodríguez
ISBN:978-84-693-7671-3/DL:T-1750-2010

2.1.1 Statistical Methods

Statistical methods are one of the early approaches proposed for texture feature extraction. They are based on the spatial distribution of gray values inside a textured image. They can be divided into three broad families: first-order statistics, second-order statistics and higher-order statistics.

2.1.1.1 First-Order Statistics

First-order statistics measure the likelihood of observing a gray value at a randomly-chosen location in the image. They only depend on individual pixel values. Mean, variance, skewness and kurtosis are some examples of this kind of methods. First-order statistics have been used in pioneering work (Weszka et al., 1976; Lumia et al., 1983), as well as in recent work (Rodríguez-Damián et al., 2006) for texture segmentation and classification.

2.1.1.2 Second-Order Statistics

Second-order statistics are defined as the likelihood of observing a pair of gray values occurring at the endpoints of a dipole of random length placed at a random location and orientation in the image. These are properties of pairs of pixel values. Some approaches belonging to this category are described below.

Given an image with its gray values quantized to a certain number of levels, Haralick et al. (1973) defined the gray-level co-occurrence matrix (GLCM), which counts the frequency with which a certain pair of gray values appears at the same relative position. A set of 28 textural features (e.g., energy, entropy, contrast, homogeneity, correlation, etc.) can be extracted from the GLCM. A similar approach is the gray-level difference method (Weszka et al., 1976), which is based on the occurrence of two pixels with a given absolute difference in gray level, separated by a specific displacement. Later, in order to bypass the intermediate step of calculating co-occurrence

matrices at various displacements, Wu and Chen (1992) proposed the statistical feature matrix, in which each element represents a feature directly extracted from the image.

2.1.1.3 Higher-Order Statistics

Higher-order statistic methods compute texture measures by considering the interaction of more than two pixels. Usually, a pixel and its surrounding neighborhood is analyzed. For instance, besides the gray level of a pixel, the *local busyness*, or gray level fluctuation measured in its neighborhood, can be used for texture classification (Weszka and Rosenfeld, 1978). In (Schachter et al., 1979), local busyness was measured by the minimum total variation given a 3×3 neighborhood. Dondes and Rosenfeld (1982) studied the alternative of using a 5×5 neighborhood, in addition to smoothing local busyness values and applying probabilistic relaxation in order to reduce variability.

Another method, the autocorrelation function (Haralick, 1979), is a feature that informs about the size of the tonal primitives on an image. If the texture is coarse, then the autocorrelation function drops off slowly. Otherwise, it drops off very rapidly. For regular textures, the autocorrelation function exhibits peaks and valleys. Features derived from the autocorrelation function were used in (Unser and de Coulon, 1982).

He and Wang (1990) introduced the texture spectrum (TS), whose basic idea is that a textured image can be decomposed into texture units (TUs) that characterize the local texture of a given pixel and its 3×3 neighborhood. In consequence, the TS is defined as the occurrence distribution of all TUs within the image. A recent attempt to combine TS and GLCM through the cross-diagonal texture matrix is described in (Al-Janobi, 2001).

The local binary pattern (LBP), which is a two-level version of the TS, was proposed in (Ojala et al., 1996). In essence, it is a binary code that describes the local texture pattern built by thresholding a neighborhood of 3×3 pixels by the gray value

UNIVERSITAT ROVIRA I VIRGILI
SUPERVISED AND UNSUPERVISED SEGMENTATION OF TEXTURED IMAGES BY EFFICIENT MULTI-LEVEL PATTERN CLASSIFICATION
Jaime Christian Meléndez Rodríguez
ISBN:978-84-693-7671-3/DL:T-1750-2010

of its center. Further developments have led to multiresolution and rotation invariant versions (Ojala et al., 2002) and to the definition of dominant local binary patterns (Liao et al., 2009).

2.1.2 Signal Processing Methods

Signal processing methods compute texture features from filtered images (in the spatial or frequency domain), which are then used in either classification or segmentation tasks. Several of these methods are compared in (Randen and Husøy, 1999).

2.1.2.1 Spatial Domain Filters

Spatial domain filters work by performing convolution operations that aim at identifying certain salient features, usually expressed in terms of *texture energy*.

The Laws (1980) masks is one of the early attempts in spatial filtering. They are derived from three simple vectors of length 3 that represent operations such as center weighting local averaging, first differencing and second differencing. By convolving these vectors with themselves or with each other, a number of zero sum 3×3 or 5×5 masks is obtained. After convolving the original image with these masks, texture energy measures can be computed.

You and Cohen (1993) realized a number of potential limitations of the Laws masks, such as not being able to capture various texture features and not being able to provide invariance due to their fixed elements. To overcome these problems, they proposed a 5×5 adaptive convolution mask tuned to be both discriminatory between different textures and invariant to rotation and scale changes. The texture energy resulting after convolution with the mask was used for classification.

Jain and Karu (1996) introduced a neural network for designing texture discrimination masks. The particularity of this network is the convolution steps taken by the hidden layers, which produce a set of feature planes. Kim et al. (2002) investigated

UNIVERSITAT ROVIRA I VIRGILI
SUPERVISED AND UNSUPERVISED SEGMENTATION OF TEXTURED IMAGES BY EFFICIENT MULTI-LEVEL PATTERN CLASSIFICATION
Jaime Christian Meléndez Rodríguez
ISBN:978-84-693-7671-3/DL:T-1750-2010

the application of support vector machines by realizing that they can incorporate feature extraction methods within their own architecture. The support vector machines received the raw gray-level values of pixels and made a first classification. Then, a neural network performed the final decision.

2.1.2.2 Frequency Domain Filters

The basic assumption of these methods is that the energy distribution in the frequency domain can be useful to identify texture. Thus, if the frequency spectrum of texture images is decomposed into a sufficient number of subbands, the spectral energy characteristics among them will be different. This proposition has led to the *multichannel filtering paradigm*, which has been extensively used. Some major approaches are reviewed in the following paragraphs.

Fourier Transform-Based Methods

Early studies based on the Fourier transform relayed on energy measures derived from the power spectrum extracted from ring-shaped and wedge-shaped regions (e.g., Bajcsy, 1973; Weszka et al., 1976). These approaches correspond to the observation that the radial distribution of values of the Fourier spectrum is sensitive to texture coarseness, whereas the angular distribution of the same values is sensitive to texture directionality. These characteristics have also been exploited in posterior work (e.g., Tsai and Tseng, 1998; Tsai and Hsieh, 1999; Lee and Chen, 2005).

Some problems with the Fourier representation have also been realized, as it is the fact that the two-dimensional Fourier transform is only a valid representation for periodic images. When this condition is not satisfied, spurious spatial frequencies appear along the axes of the transformed plane, which is known as *aperture effects* (Dyer and Rosenfeld, 1976). To avoid this problem, the two-dimensional cosine transform may be utilized instead, which is equivalent to the two-dimensional Fourier transform of a symmetric image.

UNIVERSITAT ROVIRA I VIRGILI
SUPERVISED AND UNSUPERVISED SEGMENTATION OF TEXTURED IMAGES BY EFFICIENT MULTI-LEVEL PATTERN CLASSIFICATION
Jaime Christian Meléndez Rodríguez
ISBN:978-84-693-7671-3/DL:T-1750-2010

Another problem is that the Fourier transform describes the frequency content of an image without any reference to localization in space. Spatial dependency can be achieved by incorporating a window function, resulting in the short-time Fourier transform (Portnoff, 1980). Another approach to achieve localization is the multiresolution Fourier transform (Wilson et al., 1992), which attempts to unify both Fourier and wavelet analysis, and which has successfully been applied to texture segmentation in (Hsu et al., 2000).

Wavelet Transform-Based Methods

The features of an image generally characterize different physical structures of the scene when perceived at different resolutions. One may start by analyzing a given image at a coarse resolution, thus getting information about the context and then proceed towards finer resolutions to get more specific details. This is precisely what Mallat (1989) showed by describing a mathematical model for the concept of *multiresolution representation* based on the wavelet transform.

The decomposition of the original image through the wavelet transform can be interpreted as a decomposition in a set of independent frequency channels, which is closely related to Marr's human vision model (Marr, 1982). The wavelet transform can also be viewed as a natural evolution of the short time Fourier transform, as it utilizes a window size which varies with frequency, thus providing a better characterization in the spatial and frequency domains. However, despite its obvious advantages, the standard wavelet transform has three limitations: it is not suitable for the analysis of high-frequency signals with relatively narrow bandwidth, it is shift sensitive and it has poor directional selectivity.

To overcome the first problem, Chang and Kuo (1993) and Laine and Fan (1993) proposed a tree-structured wavelet transform (wavelet packet), which differs from the standard transform in that both the high- and low-frequency bands are decomposed at each iteration. Later, M-band wavelets (Chitre and Dhawan, 1999), a multiband

generalization of the two-band transform, were also studied.

Shift insensitivity was achieved by an overcomplete wavelet decomposition called discrete wavelet frame (Unser, 1995; Laine and Fan, 1996), where the output of the filter banks is not subsampled. An extension that combines this and the previous M-band approaches is the M-band wavelet packet frame described in (Acharyya et al., 2003).

Solutions for the third limitation include hexagonal quadrature mirror filters (Simoncelli and Adelson, 1990), steerable filters and their multi-scale extension: the steerable pyramid (Freeman and Adelson, 1991), modulated wavelets (Hsin and Li, 1998), curvelets (Candès and Donoho, 2000), the complex wavelet transform (Hatipoglu et al., 2000), rotated wavelet filters (Kim and Udpa, 2000) and contourlets (Po and Do, 2006), among others.

Gabor Filter-Based Methods

A two-dimensional Gabor function is a complex sinusoidal grating modulated by a two-dimensional Gaussian in the space domain, which corresponds to a shifted Gaussian in the spatial frequency domain. Thus, it can be thought of as a local band-pass filter. It is possible to configure Gabor filters to have various shapes, bandwidths, center frequencies and orientations by varying a number of related parameters. Therefore, they are generally used as a filter bank that covers specific regions of the frequency plane.

The main reason that makes Gabor filters appealing for signal processing tasks is that Gabor functions are able to minimize the uncertainty relationship between the time (or space) and frequency domains as proved by Gabor (1946) for the one-dimensional case and, later, by Daugman (1985) for the two-dimensional case. Early work by Clark and Bovik (1986) and Turner (1986) has shown their usefulness for texture analysis.

Paradoxically, the parameter flexibility mentioned above is possibly the main diffi-

culty when designing a filter bank. The effects of different filter parameters on texture classification have been studied in (Chen et al., 2004; Bianconi and Fernández, 2007). The vague and empirical nature of parameter selection has motivated explicit optimization procedures. Teuner et al. (1995) proposed parameter tuning based on detection of nonconforming components computed from a spectral contrast function. In (Dunn and Higgins, 1995), filters to segment pairs of textures are selected according to a rank ordering obtained from computing the misclassification probability. Extensions to this work to reduce the segmentation error through Gaussian post-filtering and to segment multi-textured images were presented in (Weldon et al., 1996) and (Weldon and Higgins, 1996), respectively. Manjunath and Ma (1996) described a strategy that made the half-peak magnitude support of the filter responses in the frequency spectrum touch each other, thus reducing redundancy. In (Tsai et al., 2001), a stochastic search based on simulated annealing was applied in order to design a single Gabor filter for multiple texture segmentation. Bresch (2002) proposed to maximize the variances of the frequency components and the spar hyper volume spanned by normalized feature vectors.

2.1.3 Structural Methods

Structural methods regard texture as being composed of texture elements or primitives. A primitive is a connected set of resolution cells characterized by a list of attributes. There exists a number of neighborhood operators to extract primitives as those utilized in edge detection, adaptive region extraction, mathematical morphology, etc. Once the primitives and information about them have been identified, there are two major approaches in order to analyze texture: computing statistical properties from the primitives and using these measures as features, which yields a *statistical approach based on structure*, or trying to determine the placement rule that describes texture, which yields *pure structural methods*.

2.1.3.1 Statistical Approaches Based on Structure

Tsuji and Tomita (1973) first segmented a scene into atomic regions based on gray tone similarity. Associated with these primitives, there was a list of attributes such as size and shape. A histogram of these properties was computed and the primitives were labeled according to the modes of the histogram.

The gray-level run-length method (Galloway, 1975) is based on computing matrices according to the number of gray-level runs of various lengths at a specific orientation present in an image. A gray-level run is a set of consecutive and collinear pixels having the same gray-level value. Its length is the number of pixels in it. A development of new run-length matrices based on a multi-level dominant eigenvector estimation algorithm can be found in (Tang, 1998).

Tuceryan and Jain (1990) proposed the extraction of texture tokens by using the properties of the Voronoi tessellation of the image. For texture analysis, features of each Voronoi cell were extracted and tokens with similar features were grouped to form uniform texture regions.

The statistical geometrical features (Chen et al., 1995) are based on geometrical properties of regions in a stack of binary images. For each binary image, attributes such as the number of connected regions and their irregularity are considered. The complexity curve method (Baheerathan et al., 1999) computes the number of black-to-white boundaries observed in binary images resulting from thresholding their gray-level original image with all possible threshold values.

2.1.3.2 Pure Structural Methods

Carlucci (1972) suggested a texture model using line segments and polygons with their placement rules expressed syntactically in a graph-like language. Zucker (1976) proposed a method in which real textures are regarded as distorted versions of ideal ones. The underlying ideal texture is represented by means of a regular graph in

which each node is connected to its neighbors in an identical fashion. The graph is then transformed by distorting the primitive at each node in order to generate a real texture.

In (Lu and Fu, 1978), the placement rule is modeled by a tree grammar. The textured image is divided into windows and the spatial structure of the cells inside the windows is expressed as a tree. The assignment of gray tones to the cells is given by the rules of the grammar. Jayaramamurthy (1979) utilized a multi-level array grammar for encoding textural patterns such that the terminal symbols of one layer are the starting symbols of the next lower one.

In (Sánchez and Lladós, 2001), a graph grammar to model textured symbols in a graphics recognition framework is presented. It follows a region adjacency graph representation of a vectorized document. Productions are based on the neighboring relations of the shapes forming the textured symbols.

2.1.4 Model-Based Methods

Model-based methods construct a parametric generative model of the observed intensity distribution, which is considered to be a combination of a function representing the known structural information on the image and a random noise sequence. The key problem is how to estimate this function and how to choose an appropriate model. Once the model is complete, it can also be used to synthesize texture.

2.1.4.1 Random Mosaic Models

Mosaic models are a type of region-based, generative models that use random geometric processes in the plane to provide image structure. There are two main classes of mosaic models: cell structure models and coverage models (or "bombing" models). Cell structure models tessellate a planar region into cells and assign a color to each cell according to a fixed set of probabilities. Coverage models are obtained by a ran-

dom arrangement of a set of geometric figures ("bombs") through a point dropping process. A discussion of different aspects of mosaic models applied to image and texture modeling can be found in (Ahuja and Rosenfeld, 1981).

2.1.4.2 Autoregressive Models

Autoregressive (AR) models, which are called simultaneous autoregressive (SAR) models in the bidimensional case, assume that each pixel in an image can be expressed as a linear combination of the neighboring pixels and an additive noise field.

Since first applied by McCormick and Jayaramamurthy (1974) for texture synthesis, the early autoregressive methods have experimented several improvements. Kashyap (1984) introduced a general class of two-dimensional ARMA models. Kashyap and Khotanzad (1986) developed a circular symmetric AR (CSAR) model for invariant texture analysis. Mao and Jain (1992) proposed a multivariate rotation invariant SAR (RISAR) and then extended it to a multiresolution SAR (MR-RISAR) model. Bennett and Khotanzad (1998) developed a multispectral extension of the gray-level SAR model and applied it to color texture images. In (Comer and Delp, 1999), a multiresolution Gaussian AR (MGAR) model for constructing a pyramidal representation of images was presented. In (Ilow and Leung, 2001), a two-dimensional fractionally integrated ARMA (FARIMA) model was utilized as texture model for synthetic aperture radar images.

2.1.4.3 Markov Random Field Models

A family of random variables is said to be a Markov random field (MRF) on a given set of sites with respect to a neighborhood system, if and only if, the *Markovianity* condition is satisfied. Markovianity depicts the local characteristics of the random field in the sense that the probability that a site is in a given state is entirely determined by the probabilities of the states of the neighboring sites.

Several MRF models of texture have been proposed. Auto-models are a kind

UNIVERSITAT ROVIRA I VIRGILI
SUPERVISED AND UNSUPERVISED SEGMENTATION OF TEXTURED IMAGES BY EFFICIENT MULTI-LEVEL PATTERN CLASSIFICATION
Jaime Christian Meléndez Rodríguez
ISBN:978-84-693-7671-3/DL:T-1750-2010

of MRFs with special constraints on two labels. Auto-models can be auto-binomial (Cross and Jain, 1983), if they follow a binomial distribution; auto-normal or Gaussian (Chellappa and Chatterjee, 1985), if the joint distribution is normal; or auto-logistic, if the label set is discrete and binary, which can be extended to a multi-level logistic model (Derin et al., 1984) if a larger number of labels is used.

Other useful MRF models are reviewed next. Multiresolution MRFs first process an image at a coarse resolution and then proceed to finer resolutions (Bouman and Liu, 1991; Krishnamachari and Chellappa, 1997; Noda et al., 2002). The hierarchical model (Won and Derin, 1992) is a two-level model for segmentation in which region formation is modeled by the high level MRF and regions are filled in by patterns generated according to MRFs at the low level. Multispectral MRFs (Bennett and Khotanzad, 1999) allow for the representation of textured color images. Partially ordered Markov models (Davidson et al., 1999) can be used to explicitly describe the joint probability. The strong MRF model (Paget, 2004) can bypass the need for maximum likelihood estimation.

2.1.4.4 Fractal-Based Methods

Fractals correspond to the idea that a given object, especially a textured area, can be represented by similar characteristics repeated at different scales. Thus, this property, known as *self-similarity*, can explicitly encode the scaling behavior of texture. In addition, the *fractal dimension* (FD) has been shown to be highly correlated with the human perception of roughness (Pentland, 1984). However, as most natural textures are not deterministic but show statistical variations, it is difficult to estimate the FD.

A number of methods has been proposed for estimating the FD of an image. They are based on ϵ-blankets (Peleg et al., 1984), the fractional Brownian function (Peleg et al., 1984), box counting (Gangepain and Roques-Carmes, 1986; Keller et al., 1989; Sarkar and Chaudhuri, 1994), maximum likelihood estimation (Lundahl et al., 1986), wavelet analysis (Wornell and Oppenheim, 1992; Liu et al., 2000), power spec-

tral density of Wold decomposition (Liu and Chang, 1997), minimum cluster volume clustering (Tolle et al., 2003), etc.

Since the FD alone may not suffice to classify or segment all real textures, as different textures may have the same FD, other related features such as multifractals (Chaudhuri and Sarkar, 1995; Xia et al., 2006a), generalized Hurst parameters and multi-scale roughness (Kaplan and Kuo, 1995; Charalampidis and Kasparis, 2002), and lacunarity (Du and Yeo, 2002), have also been proposed.

2.2 Optimal Evaluation Window for Texture Segmentation

Most of the algorithms for texture classification and segmentation proposed so far assume that rectangular (square) samples of significant size of each particular texture are available. In practice, this may not be the case, since irregular shapes of different sizes are more likely to be found in real images. Therefore, both the shape and size of the evaluation window used during texture feature extraction should be carefully selected.

In this sense, the study in (Garcia-Sevilla and Petrou, 2000) is quite revealing, since it concludes that texture characterization is much more influenced by the window size than by the window shape. However, in the vast majority of cases, the optimal evaluation window size is set on an empirical basis. Moreover, cues about the value of this parameter are only given after validating the proposed classification or segmentation technique on the test set (e.g., Jain and Karu, 1996; Kim et al., 2002; Muneeswaran et al., 2006; Kim and Kang, 2007; Yang et al., 2008).

There have been very limited attempts to systematically determine the optimal evaluation window size for texture segmentation. For instance, Puig and Garcia (2001) proposed an algorithm for determining the smallest window size that maximizes the discrimination capabilities of a given texture method for a given set of

texture patterns. The discrimination power is measured by the amount of overlap between pairs of discrete probability density functions obtained after applying the evaluated method to each pair of texture samples. If the area of the non-overlapping portion is above a certain threshold, a vote for that window size is registered. In the end, normalized votes and discrimination percentages are averaged in order to obtain the final selection.

Unfortunately, further experiments in (Puig and Garcia, 2003) shown that there is a large variability on the optimal window size, even for the same texture methods, depending on the texture patterns to be classified. Moreover, they shown that a window size considered to be optimal from a theoretical standpoint does not necessarily lead to optimal classification/segmentation results given an arbitrary test image.

Motivated by this observation, Puig and Garcia (2003) proposed to integrate features obtained after the evaluation of texture methods over multiple windows of different size using a linear opinion pool where the weights are indicators of the discrimination power of each window size. Weights are computed in a similar way as in (Puig and Garcia, 2001), but using the Kullback J-divergence as a separability measure. Further refinements were later introduced in (Puig and Garcia, 2006), with the inclusion of a conflict resolution mechanism that improves the weighting scheme by dealing with contradictory opinions.

A simpler approach is followed in Melendez et al. (2008), where multiple evaluation window sizes are directly fused by the KNN rule. In this case, a set of prototypes is determined for each texture pattern considering each evaluation window size. Then, the voting scheme of the KNN classifier is the one that integrates the different information sources. More recently, Rao et al. (2009) proposed to use a hierarchy of window sizes to deal with degenerate regions in their segmentation algorithm. Starting from the largest window size, they recursively apply an agglomerative process with ever smaller window sizes till all degenerate regions have been merged with their adjacent ones. They utilized 7×7, 5×5, 3×3 and 1×1 window sizes.

Finally, a different although somehow related approach is to increase (or decrease) the spatial extent of texture operators, thus achieving a multiresolution scheme. One of the most representative cases is reported in (Ojala et al., 2002), where LBP operators with different radius (1, 2 and 3 pixels) are separately applied and then combined by defining an aggregate dissimilarity measure as the sum of individual likelihoods computed from the responses of individual operators. Another relevant approach in this line is the MSAR computation proposed in (Mao and Jain, 1992), where the neighborhood set changes to achieve the effect of multiresolution, leaving the image unchanged (neither subsampling nor low-pass filtering are involved).

2.3 Supervised Texture Classification

Many pattern recognition techniques (e.g., Duda et al., 2001; Bishop, 2006) that use textural information to identify the different classes present in an image are utilized for supervised texture classification. Any of these techniques have the potential to be extended to supervised pixel-based texture classification. The main approaches among them are summarized in this section. In addition, in order to improve the efficiency of a classifier, feature selection (Guyon and Elisseeff, 2003; Liu and Yu, 2005; Saeys et al., 2007) may be applied in order to determine the most useful texture features. The main aspects related to this topic are also reviewed below.

2.3.1 Supervised Classification Methods

Supervised classification methods can be roughly divided into three groups: decision theory methods, syntactic methods and structural methods.

2.3.1.1 Decision Theory Methods

These methods use *decision functions* (or *discriminant functions*) in order to classify an input pattern. The objective is to determine as many decision functions as neces-

sary such that, if an input pattern belongs to a specific class, the decision function computed for that class evaluated on that pattern must produce a higher value than the functions corresponding to the other classes. A wide variety of decision methods have been reported in the literature. The differences among them are due to the way decision functions are defined. The most representative methods within this group are summarized below.

Template Matching

In this case, every class is represented by a feature vector considered to be an ideal *prototype* or *template*. Then, in order to classify an input vector, a distance function from this vector to each of the prototypes is measured and the input vector is assigned to the class with the nearest prototype. Several distance functions have been proposed, such as the Manhattan or the Euclidean distances (which are particular cases of the Minkowski metric), the Tanimoto metric, the Mahalanobis distance, etc. The selection among these or other distance functions is usually dictated by computational concerns.

In practice, this type of classifier works well when the distance between means is large compared to the spread of each class and distributions are spherical (or hyperspherical). Some recent works based on this classification method are reported in (Kurmyshev and Sánchez-Yáñez, 2005; Cui et al., 2006; Choy and Tong, 2008).

K-Nearest Neighbors

The k-nearest neighbors (KNN) rule classifies an input vector by assigning it to the class most frequently represented among the vector's nearest samples. Some positive aspects of the KNN classifier are that it does not make any assumptions about the underlying densities and that its error rate tends to the Bayes optimum as the number of neighbors increases. In practice, however, the dense sampling required by this asymptotic guarantee is not present. Thus, it may suffer from variation in terms of bias-variance. Some works based on this method are presented in (Singh et al., 2001;

UNIVERSITAT ROVIRA I VIRGILI
SUPERVISED AND UNSUPERVISED SEGMENTATION OF TEXTURED IMAGES BY EFFICIENT MULTI-LEVEL PATTERN CLASSIFICATION
Jaime Christian Meléndez Rodríguez
ISBN:978-84-693-7671-3/DL:T-1750-2010

Zhang et al., 2006).

A key point for achieving a good classification result with the KNN rule is to appropriately determine the most representative samples or prototypes that characterize each of the considered patterns. In the context of texture classification, these prototypes are usually referred to as *textons* according to the redefinition of the term given by Malik et al. (1999), and correspond to the centroids of clustered feature vectors obtained after evaluating several texture feature extraction methods. In most cases, textons are computed through classical approaches, such as learning vector quantization (Ojala et al., 2001) and k-means (Malik et al., 1999; Leung and Malik, 2001; Varma and Zisserman, 2005; Yang et al., 2007; Kim and Hong, 2009), or by more recent clustering algorithms, such as mean shift (Cula and Dana, 2001; Georgescu et al., 2003). In general, the number of desired textons (or a parameter related to it) is set by the user.

In addition, in the related field of image retrieval and categorization, there is a recent trend about the use of image characteristic points, which are salient image patches containing rich local information about the scene. These keypoints are then grouped into a large number of clusters whose centroids are called *visual words* in analogy to the *word* representation used for text categorization, and whose role is similar to the one played by textons. Therefore, as with textons, most approaches have relied on general purpose clustering algorithms, such as k-means (Sivic and Zisserman, 2003; Winn et al., 2005; Nister and Stewenius, 2006; Yang et al., 2007), for generating the visual word vocabulary. However, since those approaches do not typically lead to a very compact and effective set, specific algorithms have also been proposed (e.g., Perronnin, 2008; Moosmann et al., 2008; Larlus and Jurie, 2009; Lazebnik and Raginsky, 2009).

Bayesian Decision Theory

Bayesian decision theory is a statistical approach to the problem of pattern classifica-

UNIVERSITAT ROVIRA I VIRGILI
SUPERVISED AND UNSUPERVISED SEGMENTATION OF TEXTURED IMAGES BY EFFICIENT MULTI-LEVEL PATTERN CLASSIFICATION
Jaime Christian Meléndez Rodríguez
ISBN:978-84-693-7671-3/DL:T-1750-2010

tion that is based on quantifying the tradeoffs between various classification decisions using probability and the costs that accompany such decisions. Decisions are taken based on the *a posteriori conditional probability* (or *posterior*), which represents the probability of observing a specific class provided that a feature (or feature vector) has been measured.

Some work in texture classification based on Bayesian methods is summarized next. In (Manduchi, 1999), mixture models for color and texture are computed. A similar approach is taken in (Boldys, 2003), but with a region merging process based on a region adjacency graph. Serrano et al. (2004) integrated semantic features with color and texture by means of a Bayesian network. In (Jurie and Triggs, 2005; Lazebnik et al., 2005), the naive Bayes classifier is used for scene analysis and object recognition. Weldon (2006) presented a modified Bayesian classifier that distorts the original density functions in order to improve the accuracy near boundaries of regions with different textures. Puig and Garcia (2006) proposed the integration of multiple texture methods and evaluation window sizes through a linear opinion pool that uses a set of reliability measures as weights.

Supervised Neural Networks

The main drawback of the methods discussed so far is that they cannot yield the minimum classification error for classes that are not linearly separable. Thus, non-linear functions are necessary for obtaining appropriate arbitrary decision regions. Neural networks provide such a mechanism, as they implement linear discriminants in a space where the inputs have been mapped nonlinearly. Therefore, they can learn the necessary nonlinearity at the same time as the linear discriminant.

Several neural network configurations have been proposed for texture classification, such as feedforward networks (Rohrmus, 2005), radial basis function networks (Baltzakis and Papamarkos, 2001), cellular networks (Szirányi and Csapodi, 1998), probabilistic networks (Raghu and Yegnanarayana, 1998), fuzzy ARTMAPs (Char-

alampidis et al., 2001), hybrid networks (Dokur and Ölmez, 2002), limited receptive area neural classifiers (Makeyev et al., 2008), etc.

Support Vector Machines

A support vector machine (SVM) is a binary classifier that represents input vectors in a high-dimensional space (typically much higher than the original space) through an appropriate nonlinear mapping, where data from two classes can "always" be separated by a hyperplane. In particular, SVMs determine the hyperplane for which the *margin*, which is defined as the smallest distance between the decision boundary and any of the samples, is maximized. Since SVMs simultaneously minimize the empirical classification error and maximize the geometric margin, they are also known as *maximum margin classifiers*.

Several texture classification schemes based on binary SVMs have been reported. They follow combination strategies such as one-against-others (Hernández et al., 2007), one-against-one (Gelzinis et al., 2007) or all-at-once (Rajpoot and Rajpoot, 2004). In addition to the commonly used linear, polynomial or radial basis kernels (Kumar and Zhang, 2006; Park and Park, 2007), specific, texture-oriented kernels (e.g., Sabri and Alirezaie, 2004; Kameyama and Taga, 2004; Horikawa, 2004) have been utilized for texture classification.

Hidden Markov Models

A hidden Markov model (HMM) is a statistical model in which the system being analyzed is assumed to be a Markov process (i.e., the conditional probability distribution of future states only depends on the present state) with unknown parameters. Thus, the objective is to determine the hidden parameters from the observable ones. The extracted model parameters can then be used to perform further analysis.

HMMs were used by Chen and Kundu (1994) in order to exploit the dependence among image subbands after quadrature mirror filtering. Wu and Wei (1996) converted 2-D texture images to 1-D signals by spiral resampling and then modeled

UNIVERSITAT ROVIRA I VIRGILI
SUPERVISED AND UNSUPERVISED SEGMENTATION OF TEXTURED IMAGES BY EFFICIENT MULTI-LEVEL PATTERN CLASSIFICATION
Jaime Christian Meléndez Rodríguez
ISBN:978-84-693-7671-3/DL:T-1750-2010

the extracted features as an HMM. In (Chen, 1997), stationary first order HMMs were used in combination with feature vectors extracted through sequentially rotated macro-masks. Choi and Baraniuk (2001) introduced a wavelet-domain hidden Markov tree (WDHMT) as the basis for multi-scale image segmentation. Improvements to this approach were presented in (Do and Vetterli, 2002) through steerable and rotation-invariant models; in (Fan and Xia, 2003), by exploiting the cross correlation across wavelet subbands; and in (Dasgupta and Carin, 2006), where a variational Bayes formulation was proposed.

Decision Trees

A decision tree is a way to represent a sequence of questions asked in order to classify a given pattern, where the first (or *root*) node is connected to other nodes by successive (directional) links (or *branches*), and these nodes are similarly connected until terminal (or *leaf*) nodes are reached, which have no further links. Generic methods for creating decision trees are: CART, ID3, C4.5, etc.

The classification process begins at the root node, which asks the value of a particular property. Based on the answer, the appropriate link to a descending node is followed. At that node, a new decision is made and, again, the appropriate link is followed. This decision making procedure continues until a leaf node is reached, where the analyzed pattern is assigned a category label. Some representative works that used decision trees for texture classification are: (Simard et al., 2000; Jalba et al., 2004; Kalinin et al., 2005).

2.3.1.2 Syntactic Methods

The idea behind syntactic pattern recognition is to define patterns in terms of *primitives* and a group of rules that determines the interactions between them, known as *grammar*, and then to use a *parsing* process for recognition.

Grammars can be unidimensional (string grammars), which are formally defined

UNIVERSITAT ROVIRA I VIRGILI
SUPERVISED AND UNSUPERVISED SEGMENTATION OF TEXTURED IMAGES BY EFFICIENT MULTI-LEVEL PATTERN CLASSIFICATION
Jaime Christian Meléndez Rodríguez
ISBN:978-84-693-7671-3/DL:T-1750-2010

by four components: a finite group of primitive symbols, a finite group of intermediate symbols, a root symbol and a set of production rules; or multidimensional (tree grammars), in which, production rules represent interconnections between elements of the tree and, in addition, there is a fifth component corresponding to the number of direct descending nodes from a terminal node. On the other hand, according to the type of structure of the productions, grammars can be free (or unrestricted), context-sensitive, context-free or regular (finite state grammars).

Given a set of grammars representing different models or classes, a test pattern is classified according to which grammar could have produced it by following a process called parsing, which consists of finding a derivation in the grammar that leads to the test pattern. Approaches to parsing include bottom-up and top-down algorithms, as well as model-dependent methods, such as finite-state machines. Some works considering syntactic methods for texture classification are (Vafaie and Bourbakis, 1988; Mojsilović et al., 2000; Sánchez and Lladós, 2001).

2.3.1.3 Structural Methods

Structural methods are based on explicit representation of the primitives that make up patterns through *structural prototypes* and on a recognition process based on *relational matching*. They exploit the idea of utilizing the same structure to represent both the known patterns and the unknown elements that need to be recognized. Structural methods are preferred over syntactic methods when there is not much knowledge about the structure of the primitives, the number of available patterns is small or the number of common substructures between the different patterns is also small. The two types of structures utilized by these methods are strings for unidimensional patterns, and graphs for multidimensional patterns.

During the recognition stage, direct matching between the test and training patterns using a distance function is performed. Several algorithms for analyzing the isomorphism between graphs and subgraphs have been proposed. However, their uti-

lization for image analysis has been somewhat limited. In this sense, some of the major concerns are how to reduce the search space or how to develop fault-tolerant algorithms. In the scope of texture classification, some recent works are reported in (Jung, 2001; Wu and Wang, 2006).

2.3.2 Feature Selection

As stated above, texture classification tasks start with a feature extraction stage. Since this stage is usually carried out by evaluating several texture methods, a (moderately) large number of features may need to be processed afterwards, thus reducing the overall efficiency of the utilized classification technique. In order to alleviate this problem, a preliminary feature selection stage can be applied.

In general, the goal of feature selection is to select a minimum (optimal) subset of features such that the resulting class distribution, given only that minimum subset, is equal or as close as possible to the original class distribution given all features (Koller and Sahami, 1996). In addition, in the scope of classification, the classification accuracy must not significantly decrease (Dash and Liu, 1997).

Most feature selection algorithms try to fulfill this goal by identifying *relevant features* with respect to some criterion (for a review of definitions about feature relevance, refer to Molina et al., 2002). John et al. (1994) identified three types of features according to their relevance, namely: *strongly relevant features*, which are always necessary and cannot be removed without affecting the original class distribution; *weak relevant features*, which are only necessary under certain conditions; and *irrelevant features*, which are not necessary at all. In addition, some approaches try to complement the selection process by also identifying *redundant features*. The notion of redundancy is usually associated with correlation, that is, two features are redundant to each other if their values are completely correlated.

One of the key aspects of feature selection is how to evaluate the *goodness* of a feature subset in determining an optimal one. Another key problem in feature

UNIVERSITAT ROVIRA I VIRGILI
SUPERVISED AND UNSUPERVISED SEGMENTATION OF TEXTURED IMAGES BY EFFICIENT MULTI-LEVEL PATTERN CLASSIFICATION
Jaime Christian Meléndez Rodríguez
ISBN:978-84-693-7671-3/DL:T-1750-2010

selection is how to generate (*search*) the subsets to be evaluated, since a tradeoff between result optimality and computational complexity must be reached. In the following sections, these two issues are further reviewed.

2.3.2.1 Subset Evaluation

The optimal subset is always relative to a certain evaluation function that, typically, measures the capability of a feature subset to distinguish among different class labels. In the context of classification, feature selection techniques can be organized into three categories, depending on how they combine the feature search with the construction of the classification model in order to measure the goodness of a feature subset (Guyon and Elisseeff, 2003): filter methods, wrapper methods and embedded methods.

Filter Methods

Filter methods assess the relevance of features based on the intrinsic properties of the data. In most cases, a score is calculated for each feature and low-scoring features are removed. Afterwards, the remaining features are used with the classification algorithm.

The main advantages of filter methods are that they easily scale to very high-dimensional datasets, they are computationally simple and fast, and they are independent of the classification algorithm. In consequence, feature selection needs to be performed only once, after which, different classifiers can be evaluated. This independence, however, can also be considered a disadvantage, as the filters ignore the interaction with the classifier − the search in the feature space is separated from the search in the hypothesis space. Another disadvantage is that most of the proposed filter methods are *univariate* (e.g., Efron et al., 2001; Torkkola, 2003; Fox and Dimmic, 2006), which means that each feature is evaluated separately, thereby ignoring feature dependencies, which may lead to lower classification performance compared to other techniques. To overcome this problem, a number of *multivariate* filter techniques

UNIVERSITAT ROVIRA I VIRGILI
SUPERVISED AND UNSUPERVISED SEGMENTATION OF TEXTURED IMAGES BY EFFICIENT MULTI-LEVEL PATTERN CLASSIFICATION
Jaime Christian Meléndez Rodríguez
ISBN:978-84-693-7671-3/DL:T-1750-2010

aiming at incorporating feature dependencies to some degree have been introduced (e.g., Hall, 1999; Yu and Liu, 2004; Mamitsuka, 2006).

Filter methods rely on various measures of the general characteristics of the training data. These measures can be grouped into four categories (Dash and Liu, 1997): distance measures, information measures, dependency measures and consistency measures.

Distance measures, also known as *separability measures*, are based on the assumption that instances of different classes are distant in the feature space. Therefore, it is enough to define a metric between classes and use it as a measure. The most usual distances belong to the Euclidean family. Other measures compute a probabilistic distance or *divergence* among the class-conditional probability densities. If a subset of features is good, the divergence among the conditional probabilities will be significant. On the contrary, poor features will result in very similar probabilities. Some typical divergence measures are the Chi-Squared statistic, the Kullback-Leibler divergence and the Bhattacharyya distance (Forman, 2003; Coetzee, 2005; Reyes-Aldasoro and Bhalerao, 2006).

Information or *uncertainty measures* compute the a posteriori probabilities of classes in order to determine how much information about the class of a given instance has been gained with respect to its prior probability. If all classes become roughly equally probable, the information gain is minimal and the uncertainty is maximal. Uncertainty is usually defined in terms of *entropy*. Examples of feature selection algorithms based on information gain or on entropy-based relevance are (Ben-Bassat, 1982; Singh and Provan, 1996; Wang et al., 1998).

Dependence measures quantify how strongly two variables are associated with each other, or the ability to predict the value of a variable from the value of another. The correlation coefficient (Hall, 1999) is a classical measure and can be used to find the correlation between a feature and a class. A variation of this approach consists of determining the dependence of a feature on other features, thus also assessing the

UNIVERSITAT ROVIRA I VIRGILI
SUPERVISED AND UNSUPERVISED SEGMENTATION OF TEXTURED IMAGES BY EFFICIENT MULTI-LEVEL PATTERN CLASSIFICATION
Jaime Christian Meléndez Rodríguez
ISBN:978-84-693-7671-3/DL:T-1750-2010

degree of redundancy between them (Koller and Sahami, 1996; Yu and Liu, 2004; Peng et al., 2005). The same goal can also be achieved by following the conditional mutual information maximization criterion (Fleuret, 2004). All evaluation functions based on dependence measures can be categorized as either distance or information measures (Ben-Bassat, 1982).

Consistency measures are characteristically different from the above measures due to their heavy reliance on the class information and the use of the *Min-Features bias* (Almuallim and Dietterich, 1994). These measures aim at finding the minimum number of features that separate classes as consistently as the full set of features does. An inconsistency is defined as two instances having the same feature values but different class labels. Some algorithms exploiting consistency checking have been proposed in (Schlimmer, 1993; Liu and Setiono, 1996; Liu and Motoda, 1998).

Wrapper Methods

Contrarily to filter methods, wrapper methods embed the model hypothesis search within the feature subset search. By following this setup, a search procedure in the space of possible feature subsets is defined and various subsets of features are generated and evaluated. The evaluation of a given subset is performed by training and testing a specific classification model, thus rendering this approach tailored to a specific classification algorithm. Examples of wrapper algorithms can be found in (Inza et al., 2004; Ruiz et al., 2006; Puig and Garcia, 2006).

The advantages of wrapper approaches include the ability to take into account feature dependencies and the interaction between the feature subset search and the classification model, which equals the bias of both the feature selection algorithm and the learning algorithm that will be used later on to assess the goodness of the solution. The main disadvantage of these methods is that they are computationally intensive, especially if building the classifier has a high computational cost. Another drawback is that they have a higher risk of overfitting than filter methods.

Embedded Methods

In this type of methods, the search for the optimal subset of features is built into the classifier construction. It can be seen as a search in the combined space of feature subsets and hypothesis. Therefore, similarly to wrapper approaches, embedded methods are also specific to a given classification algorithm. For instance, the goodness of a given subset can be assessed by weighted naive Bayes (Duda et al., 2001), the weight vector of SVMs (Weston et al., 2003) or the weights of logistic regression (Ma and Huang, 2005).

The main advantage of embedded methods is that they include the interaction with the classification model, while at the same time being less computationally intensive than wrapper methods.

2.3.2.2 Subset Generation

Subset generation is essentially a search process, where each state in the search space specifies a candidate feature subset for evaluation. The two basic issues about this process are: deciding the starting point (e.g., an empty set, the full set or a random subset) and deciding the search strategy. According to (Molina et al., 2002; Liu and Yu, 2005), there are three different search strategies for generating feature subsets: complete, heuristic and random.

Complete Search

Complete search guarantees that the optimal result with respect to the utilized evaluation criterion will be found. While an exhaustive search is complete (i.e., no optimal subset is missed), a complete search does not have to be exhaustive (Schlimmer, 1993). If the evaluation measure is *monotonic*, it is possible to find an optimal subset without exhaustively evaluating all subsets, by following heuristic algorithms such as branch and bound (Narendra and Fukunaga, 1977), best first (Xu et al., 1988), beam search (Doak, 1992), among others. Thus, although the order of the search space is

exponential, a smaller number of subsets is evaluated.

Sequential Search

This type of search selects one among all the successors of the current state, which is done iteratively. Once the following state is selected, it is not possible to move backwards. Consequently, these methods do not guarantee an optimal result, since the optimal solution could be in a region of space that has not been visited. Algorithms with sequential search are fast and simple to implement. Their complexity is determined by taking into account the number of evaluated subsets in each state change. Therefore, their cost is polynomial (usually quadratic or less). Examples of sequential search are sequential forward generation, sequential backward elimination and bidirectional selection (Liu and Motoda, 1998).

Random Search

Random search strategies start with a randomly selected subset and then proceed in two different ways. One way is to follow sequential search, thus providing randomness to the above approaches. Examples are random start hill-climbing and simulated annealing (Doak, 1992). The other way is to generate the next subset in a completely random manner, that is, a current subset does not grow or shrink from any previous subset according to a deterministic rule. Examples are genetic algorithms (Vafaie and Imam, 1994) and Las Vegas-type algorithms (Liu and Setiono, 1996).

2.4 Unsupervised Texture Segmentation

Image segmentation methods can be classified into: boundary-based methods, region-based methods and the combination of both. In the context of texture analysis, segmentation methods are usually unsupervised. However, there have been some attempts to combine unsupervised segmentation and supervised classification methods by following two-stage strategies where a supervised classifier is used to refine the

UNIVERSITAT ROVIRA I VIRGILI
SUPERVISED AND UNSUPERVISED SEGMENTATION OF TEXTURED IMAGES BY EFFICIENT MULTI-LEVEL PATTERN CLASSIFICATION
Jaime Christian Meléndez Rodríguez
ISBN:978-84-693-7671-3/DL:T-1750-2010

previous unsupervised segmentation of a given image.

2.4.1 Boundary-Based Methods

Boundary-based (or edge-based) methods rely on the generation of a strength image and the extraction of prominent edges. In other words, they are based on some discontinuity property of image pixels.

Some segmentation approaches have utilized local filtering techniques, such as edge detection operators, in order to find "textural edges", which are defined as boundaries of homogeneously textured regions located where sudden changes in local characteristics occur (Thompson, 1977; Khotanzad and Chen, 2002).

A closely related group of methods have considered the idea of computing the gradient of the texture feature space. In this way, it is possible to associate texture boundaries with local peaks of the texture gradient magnitude (Malik and Perona, 1990; Manjunath and Chellappa, 1993) or to build edge flow vectors derived from the magnitude of the gradient and predicted errors (Ma and Manjunath, 2000).

Alternatively, edge detection can be performed in the opposite way, that is, by starting with a continuous curve model and then trying to localize it on the maxima of the gradient through deformations of the original contour. These active contour models (or "snakes") are energy-minimizing curves governed by internal forces that impose the mechanical properties of the model (elasticity and rigidity), as well as by external forces that push the curve towards the desired boundaries or that impose constraints defined by the user. Some works that utilize edge-based active contours for texture segmentation are: (Sagiv et al., 2000; Ray et al., 2001; Alemán-Flores et al., 2007).

Another edge detection mechanism is anisotropic diffusion (Perona and Malik, 1990), which is a multi-scale edge-preserving technique that encourages intra-region smoothing versus inter-region smoothing. In (Rubner and Tomasi, 1996), this idea was extended to feature vectors that describe textural content. Posterior research has

led to a more efficient implementation known as stabilized inverse diffusion equations, which have been extended to vector-valued images by Dong and Pollak (2006).

2.4.2 Region-Based Methods

Region-based methods rely on the homogeneity of spatially localized features and properties of pixels in relation to their local neighborhood. In this sense, segmentation can be defined as a constrained partitioning problem, where each partitioned region should be as homogeneous as possible and neighboring ones should be as different as possible. Region-based methods can be roughly classified into three categories: clustering methods, random field approaches and region-based active contours.

2.4.2.1 Clustering Methods

The objective of these methods is to discover sets or *clusters* of patterns in the feature space whose members are more similar to each other than to other patterns. Several algorithms for defining these sets have been proposed. According to them, clustering techniques may be divided into three subcategories: sequential clustering, cost minimization clustering and hierarchical clustering.

In *sequential clustering*, feature vectors are presented to the algorithm a reduced number of times (or even once). Therefore, they are usually very fast. However, the results are dependent on the order of presentation. In the context of image segmentation, one of the most representative techniques belonging to this type of clustering algorithms is region growing. In most implementations, it starts by choosing initial points called seeds and then regions grow by adding neighboring pixels according to certain homogeneity criterion (Chang and Li, 1994; Shafarenko et al., 1997; Deng and Manjunath, 2001; Scarpa and Haindl, 2006). Other relevant approaches to sequential clustering are those based on self-organizing neural networks (Mao and Jain, 1996).

Cost minimization clustering algorithms perform an iterative process that pro-

duces improved results after each step until the optimal (or a suboptimal) value for the defined cost function is reached. The procedures utilized to assign feature vectors to each cluster include: crisp, probabilistic, possibilistic and fuzzy techniques. Several texture segmentation methods based on cost minimization clustering algorithms such as k-means (Muneeswaran et al., 2006), fuzzy c-means (Xia et al., 2006a), isodata (Laine and Fan, 1996), expectation-maximization (Zöller and Buhmann, 2007), mean shift (Ozden and Polat, 2007), graph partitioning (Kim and Hong, 2009), among others, have been proposed.

Unlike previous methods, *hierarchical clustering* algorithms do not produce a "flat" data description, but a hierarchy or tree in which, at each subsequent level, there are more or fewer subclusters depending on the approach: divisive or agglomerative, respectively. Some hierarchical texture segmentation methods have been reported in (Puzicha et al., 2000; Yang et al., 2008). A combination of both agglomerative and divisive approaches result in the split and merge algorithm. For image segmentation, the split and merge technique first starts by considering the whole image as a single region. If this region does not satisfy a homogeneity criterion, it is split into a number of subregions and each of them is tested in the same way. This process is recursively repeated until every resulting region contains homogeneous pixels. In the second step, all adjacent regions with similar attributes are merged. Some works based on split and merge are: (Ojala and Pietikäinen, 1999; Huang et al., 2006).

2.4.2.2 Random Field Methods

These methods use a random field (usually an MRF) to model the segmentation map of an image. Thus, the image pixels are labeled after appropriate characterization by maximizing the a posteriori probability of this field. A particular case where both feature estimation and labeling stages are performed by mutually dependent MRFs yields the hierarchical model reviewed in Section 2.1.4.3.

UNIVERSITAT ROVIRA I VIRGILI
SUPERVISED AND UNSUPERVISED SEGMENTATION OF TEXTURED IMAGES BY EFFICIENT MULTI-LEVEL PATTERN CLASSIFICATION
Jaime Christian Meléndez Rodríguez
ISBN:978-84-693-7671-3/DL:T-1750-2010

Bouman and Liu (1991) used a combination of AR and MRF models to perform multiresolution segmentation. Another multiresolution approach based on GMRFs was presented in (Krishnamachari and Chellappa, 1997). Sarkar et al. (2002) proposed an MRF-based segmentation approach for multispectral imagery that can account for tonal and textural features. Deng and Clausi (2004) presented a scheme for combining labeling and feature components by introducing a variable weighting parameter between them. Xia et al. (2006b) alternately optimized the feature set and labeling by simulated annealing. Benboudjema and Pieczynski (2007) utilized a triplet Markov field model that allowed them to better deal with non-stationary images than the traditional MRF.

2.4.2.3 Region-Based Active Contours

Region-based active contours assume that the image consists of a finite number of regions characterized by a predetermined set of features or statistics that can be pulled apart by means of an energy functional. They were proposed in order to overcome some of the problems shown by their edge-based counterparts, such as sensitivity to both noise variability and initial contour placement.

Some representative contributions to this type of active contours are: the active regions of Ivins and Porrill (1995), the region competition framework developed by Zhu and Yuille (1996), the active contours "without edges" proposed by Chan and Vese (2001), the active polygons developed by Unal et al. (2005) and the graph partitioning active contours described in (Sumengen and Manjunath, 2007).

2.4.3 Boundary- and Region-Based Methods

Since boundary and region information can be thought of as complementary sources, there has been a significant effort to integrate them both in order to improve segmentation results. Depending on how this information is fused, different approaches

result.

Freixenet et al. (2002) identified two types of integration depending on the time of fusion: *embedded integration* and *post-processing integration*. In the first case, integration is performed through the definition of new parameters or decision criteria for segmentation. For instance, edge information could control growing of regions or could be used to guide initial seed placement. In the second case, the integration is performed after independently applying the two approaches and its objective is to refine the initial segmentation. In this sense, the dual information can be used to eliminate false boundaries or to improve the localization of inaccurate boundaries. Some examples of both types of integration are shown in (Hsu et al., 2000; Muñoz et al., 2003; Arbeláez et al., 2009; Rao et al., 2009).

In addition, several methods that incorporate boundary and region information under the variational approach have been reported. Chakraborty and Duncan (1999) combined information by means of a game-theoretic framework. Paragios and Deriche (2002) proposed the geodesic active region model. Sagiv et al. (2006) integrated geodesic and edgeless active contours. Allili and Ziou (2007) used region information based on mixtures and boundary information modeled by polarity information.

2.4.4 Two-Stage Methods

Most two-stage methods apply a supervised classifier in order to refine the segmentation produced by a previous unsupervised method, such as any of the techniques reviewed between Sections 2.4.1 and 2.4.3.

Some methods are based on the refinement of the boundary of the segmentation. For example, Ojala and Pietikäinen (1999) performed pixelwise classification of boundary pixels after segmentation with a split and merge algorithm based on LBP histograms. They utilized a discrete disc of radius equal to 11 pixels and the G statistic as a dissimilarity measure, and treated the LBP histograms of the image segments as texture models. Camilleri and Petrou (2000) refined boundaries by extending the

ideas of linear spectral unmixing, originally applied to remote sensing, to the case where the local energy of a boundary pixel is a linear combination of the local energies of the pixels at both sides of the boundary through surrounding windows that do not touch the boundary.

Other methods use the learned classes from the unsupervised classifier as seeds for growing finer regions. Mirmehdi and Petrou (2000) blurred the image according to the blurring filters that model the way humans perceive colors from different distances and clustered the highly blurred image (coarsest level) to determine "core clusters", that is, sets of pixels that can be confidently associated with the same region. Then, those clusters were used as the basis for probabilistic assignment of the remaining, non-confident pixels in the image at various smoothed levels. A similar approach was adopted in (Luo and Savakis, 2000), where "high-confidence" regions were obtained through unsupervised clustering of multiresolution SAR features. In the second stage, "low-confidence" pixels identified by multi-level morphological erosion were reclassified by a nearest neighbor classifier trained with the centroids of the high-confidence clusters, this time using wavelet features. Kim and Kang (2007) used the Kullback-Leibler divergence as homogeneity criterion in order to distinguish between "homogeneous" and "heterogeneous" blocks of a filtered image. Once homogeneous blocks were grouped by a fuzzy c-means algorithm, heterogeneous blocks were labeled at the pixel level by a Gaussian mixture-based classifier evaluated within a 13×13 local window.

UNIVERSITAT ROVIRA I VIRGILI
SUPERVISED AND UNSUPERVISED SEGMENTATION OF TEXTURED IMAGES BY EFFICIENT MULTI-LEVEL PATTERN CLASSIFICATION
Jaime Christian Meléndez Rodríguez
ISBN:978-84-693-7671-3/DL:T-1750-2010

Chapter 3

Supervised Texture Segmentation

This chapter presents a new technique for supervised texture segmentation through supervised pixel-based texture classification. It is based on an efficient multi-level pattern classification approach that utilizes texture features obtained after evaluating a set of texture methods over multiple windows of different size. The chapter is organized as follows. Section 3.1 introduces the problem of pixel-based texture classification. Section 3.2 describes the evaluation of texture methods over multiple windows of different size and the utilization of the resulting features in the proposed multi-level classification scheme. The two pixel-based classification alternatives studied within this scheme are described in Section 3.3. The first alternative, presented in Section 3.3.1, consists of a KNN classifier fed with prototypes determined either by a resolution-driven clustering algorithm or by a normalized cut clustering algorithm. The second alternative, presented in Section 3.3.2, consists of an SVM-based classifier that follows a one-against-one strategy. A method for selecting appropriate parameters for both classifiers is also given. Section 3.4 introduces some strategies that aim at improving the efficiency of the configured pixel-based classifiers. Section 3.5 describes the experimental setup, details on how the proposed technique has been configured for experimentation and shows the obtained results. The contents of the chapter are finally summarized in Section 3.6.

3.1 Introduction

The problem of *supervised pixel-based texture classification* consists of identifying the texture pattern to which every pixel of a digital image belongs (Garcia and Puig, 2003). By doing that, the image is segmented according to the sought patterns as a collateral effect. Thus, if the image contains several regions of different texture, they can be segmented and identified by applying a *supervised pixel-based texture classifier*.

There are many real-life problems for which pixel-based texture classifiers are suitable. For instance, the identification of tissues with particular characteristics in medical images, the location of specific types of soil in satellite or aerial images, or the recognition of different objects in robot exploration. These problems do not just require the segmentation of the given image into disjoint regions, but the identification of specific regions of interest. In this sense, supervised pixel-based classification subsumes segmentation.

In order to perform pixel-based texture classification, a number of texture measures are computed for every image pixel by applying a predefined set of texture methods to its neighboring pixels. Usually, these neighborhoods are square windows centered at the considered pixel. A wide variety of texture methods have been proposed in the computer vision literature (see Section 2.1). Their performance basically depends on the type of processing they apply, the texture content and their configuration parameters. While the first two issues are intrinsic to the texture method definition and the analyzed scene, respectively, and thus, cannot be altered, the selection of appropriate parameters is left to the user. One of these parameters is precisely the size of the neighborhood of pixels over which a method is applied (evaluation window). Therefore, the problem of determining proper window sizes for evaluating each method must be addressed.

In addition, due to the nature of pixel-based classification, a problem regarding the number of feature vectors generated for the training patterns arises, since a feature

vector is computed for every image pixel. For instance, if the classification technique is based on prototypes, the latter potentially generates hundreds of thousands of prototypes to compare with. However, the use of so many prototypes may not be necessary in order to achieve satisfactory classification results and, moreover, can be prohibitive in real applications in terms of both computational time and memory usage. Therefore, a mechanism for efficient learning must be devised.

The two issues discussed above are further developed in the following sections.

3.2 Classification with Multiple Evaluation Windows Through a Multi-Level Approach

Although previous works on texture segmentation have already shown the benefits of utilizing multiple evaluation window sizes (Section 2.2), which approach is the best for combining these different sources of information is still an open issue.

For instance, Puig and Garcia (2006) integrated different evaluation window sizes by assigning them a relevance opinion (weight) according to a divergence measure that indicates how well each window size separates a given training texture pattern from the others. However, as the training patterns are single-textured images, the weight assigned by following this approach is not representative of the structure of the test image, which is usually composed of multiple textures. Furthermore, this method may be biased to the largest window, since this window captures more information and, in consequence, has better capabilities in order to distinguish between texture classes.

In (Melendez et al., 2008), multiple evaluation window sizes were fused directly by the KNN rule. Although the classification accuracy improved compared to (Puig and Garcia, 2006), this methodology has not taken full advantage of the multi-sized window paradigm yet, since there is no guarantee that the most appropriate window size will receive the majority of votes. In (Rao et al., 2009), a hierarchy of evaluation

window sizes was applied in the context of an agglomerative algorithm. However, this approach only aims at merging smaller regions and is more a post-processing step than a classification strategy.

The problem of pixel-based classification with multiple windows of different size can be formulated as follows. Let $\mathbf{I}$ be a two-dimensional input image of $NR \times NC$ pixels containing several regions of uniform texture. Let $\tau_1, ..., \tau_T$ be a set of T texture models of interest. Each model τ_k, $k \in [1, T]$, is described by a sample image $\mathbf{I}_k$ (or a set of sample images) that contains a pattern of that texture. The goal of a pixel-based texture classifier is to determine if a pixel $\mathbf{I}(c_1, c_2)$, $c_1 \in [1, NR]$, $c_2 \in [1, NC]$, belongs to any of the T aforementioned texture models. If the classifier is applied to all the pixels of $\mathbf{I}$, the result is the identification (segmentation plus labeling) of the regions that belong to the given texture models.

The classification of a pixel $\mathbf{I}(c_1, c_2)$ based on textural information starts by extracting a feature vector for that pixel, $\mathbf{f}(c_1, c_2) = (f_1(c_1, c_2), ..., f_M(c_1, c_2))$. Each feature f_i, $i \in [1, M]$, is obtained by applying a texture feature extraction method μ_i, which computes a value $\mu_i(c_1, c_2)$ from the pixels contained in a neighborhood around $\mathbf{I}(c_1, c_2)$. This neighborhood is usually a square window centered at $\mathbf{I}(c_1, c_2)$. M different texture feature extraction methods are considered. Then, the feature vector obtained in that way can be classified by applying any suitable pattern recognition technique (Section 2.3).

Since, in general, no optimal window size can be determined for any texture method and application, as the outcome of those methods greatly depends on the texture patterns to which they are applied, a proper solution consists of evaluating every texture feature extraction method μ_i over W square windows, $\{w_1, ..., w_W\}$, with each window having a different size. In this work, every window w_j, $j \in [1, W]$, contains $s_j \times s_j$ pixels, with $s_j = 2^j + 1$. Only windows with size $s_j \leq 2(\min(c_1, NR + 1 - c_1, c_2, NC + 1 - c_2) - 1) + 1$ are applicable to pixel $\mathbf{I}(c_1, c_2)$, as they entirely fit into the image. Therefore, there is always a strip of pixels that belong to the boundary of

I that is not classified. Considering this scheme, every texture method μ_i generates a feature vector $\mathbf{f}_i$ with W texture features for every pixel to which the method is applied: $\mathbf{f}_i(c_1, c_2) = (f_i^1(c_1, c_2), ..., f_i^W(c_1, c_2))$. These features are fed into the chosen classifier.

Ideally, the strategy for classifying the pixels of an image using multiple evaluation window sizes should apply large windows inside regions of homogeneous texture in order to avoid noisy classified pixels and small windows near the boundaries between those regions in order to precisely define them. Unfortunately, that kind of knowledge about the structure of the image is only available after it has already been segmented. Notwithstanding, an a priori approximation of that strategy can be devised through the following steps:

1. Set the current evaluation window, w_j, to the largest one: $w_j := w_W$.
2. Select the current evaluation window, w_j, and classify the test image pixels labeled as unknown (initially, all pixels are labeled as unknown).
3. In the classified image, locate the pixels that belong to boundaries between regions of different texture and mark them as unknown, as well as their neighbors. The size of the neighborhood corresponds to the size of the window used to classify the image ($s_j \times s_j$).
4. Discard the current evaluation window: $j := j - 1$.
5. Repeat steps 2 to 4 until the smallest evaluation window, w_1, has been utilized.

This scheme, which can be thought of as a multi-level top-down approach, has been used for pixel-based classification with the proposed technique. In addition to closely approximating the aforementioned ideal strategy for using multiple evaluation window sizes, this approach avoids the classification of every image pixel with all the available windows. Thus, it leads to a lower computation time than previous approaches. Comparisons with single-window versions of the techniques developed

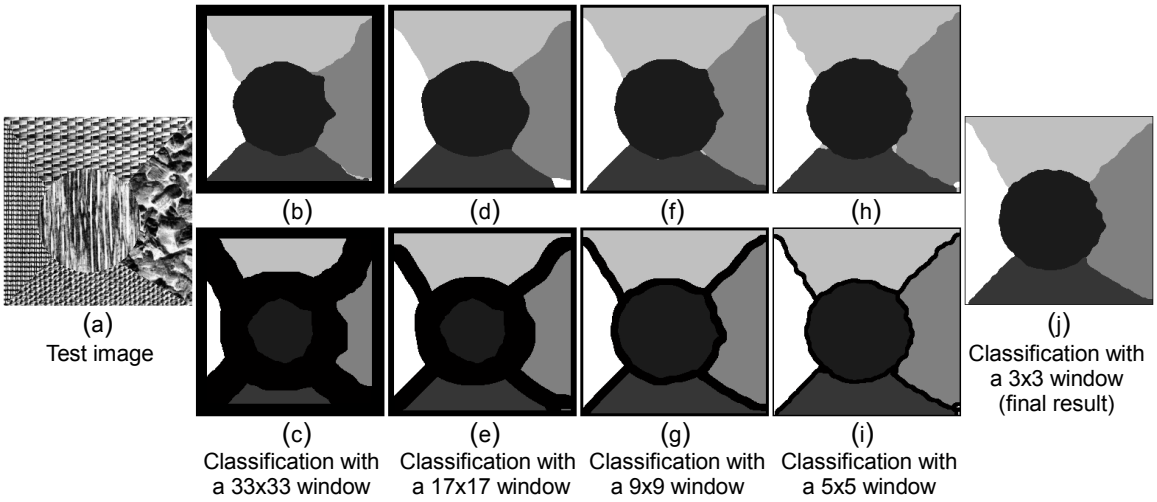

Figure 3.1: Steps performed by the proposed technique in order to classify a given test image. Boundaries between textured regions are refined after each classification level.

in this thesis are carried out in Section 3.5.4, while comparisons with alternative single-window and multi-window approaches are carried out in Section 5.2.

Figure 3.1 shows how the proposed pixel-based classification technique works when applied to a given test image, considering five evaluation window sizes ($W = 5$). First, the test image (Figure 3.1(a)) is classified with a 33×33 evaluation window, Figure 3.1(b). Then, boundaries between textured regions and their neighborhoods are labeled as unknown (black regions in Figure 3.1(c)). Next, those unknown pixels are reclassified with a 17×17 window, Figure 3.1(d), and the process is iterated until the image is reclassified with a 3×3 window, yielding the final result, Figure 3.1(j).

3.3 Supervised Pixel-Based Classification

The main concern regarding the pixel classification step (Step 2) of the multi-level scheme described in the previous section is how to appropriately use the large amount of information obtained from the training samples in order to build an accurate, yet

UNIVERSITAT ROVIRA I VIRGILI
SUPERVISED AND UNSUPERVISED SEGMENTATION OF TEXTURED IMAGES BY EFFICIENT MULTI-LEVEL PATTERN CLASSIFICATION
Jaime Christian Meléndez Rodríguez
ISBN:978-84-693-7671-3/DL:T-1750-2010

computational inexpensive classifier. In this dissertation, two techniques based on the KNN rule and SVMs are studied.

3.3.1 K-Nearest Neighbors Classifier

The KNN rule assigns a given input vector the label most frequently represented among the K nearest samples. In other words, a decision is made by examining the labels of the K nearest neighbors and choosing the one that appears the most.

Although the KNN rule is a suboptimal procedure, as it usually leads to an error rate greater than the minimum possible (the Bayes rate), it is a good choice to be used with arbitrary distributions for which the assumption that the forms of the underlying densities are known is not valid. Furthermore, even with this privileged knowledge in hand, a high-dimensional density, such as the one resulting from texture feature extraction, might be difficult to be accurately estimated due to the nature of the data. Therefore, the applicability of this type of classifier is well justified.

The KNN classifier for pixel-based classification proceeds by comparing a vector that characterizes a pixel in the test image with all the *prototypes* that model each of the T texture patterns considering the current window size. Then, a list with the K best distances among the T sets of possible values is obtained and the most voted texture among the candidates is chosen to be the texture to which the analyzed pixel belongs. If there is a tie, the winning texture is the one with the smallest distance.

A key point when training a KNN classifier is to appropriately determine the prototypes that model each of the classes to be discriminated, since they greatly influence both its accuracy and its processing time. Taking into account that after feature extraction a feature vector per image pixel is obtained and that typical sizes of the training images associated with the texture patterns are 128×128 or 256×256 pixels, it is clear that the number of vectors that could model each pattern for a given window size is enormous. Thus, it is necessary to reduce this number, which is a suitable task for a clustering algorithm, as texture prototypes can be associated

with the centroids of the resulting partition (Malik et al., 1999). The two alternatives proposed in this work, as well as an algorithm for configuring the parameters of the resulting classifier, are discussed below.

3.3.1.1 Resolution-Driven Clustering

Due to its simplicity and good convergence properties, one of the most widely used clustering algorithms for prototype selection is iterative k-means (Malik et al., 1999; Leung and Malik, 2001; Varma and Zisserman, 2005; Yang et al., 2007; Kim and Hong, 2009). However, k-means suffers from important drawbacks, such as the requirement of specifying the number of clusters beforehand, the dependence on the initial conditions that may result in suboptimal convergence, the dependence on the nature of the data, etc.

In order to overcome some of the aforementioned problems, several variations of k-means have been proposed. For instance, regarding the issue of determining the optimal number of clusters in a data set, Pelleg and Moore (2000) proposed a wrapper called x-means, which searches over a range of values according to the Bayesian information criterion or the Akaike information criterion. Later, Hamerly and Elkan (2003) also proposed a wrapper, called g-means, which determines the optimal number of clusters by continuously splitting the existing ones if they do not satisfy the Anderson-Darling statistical test for the Gaussian hypothesis. According to Hamerly and Elkan (2003), the number of clusters found by g-means is lower and more accurate than the number of clusters found by x-means. Notwithstanding, due to the search criteria they use, both x-means and g-means are clearly dependent on the shape of the clusters (e.g., Gaussian).

Alternatives to k-means have also been studied, as is the case of mean shift (Comaniciu and Meer, 2002), which, by interpreting the feature space as a probability density function, aims at locating the modes of this density and associating a cluster with each of them based on the local structure of the feature space. Contrarily

UNIVERSITAT ROVIRA I VIRGILI
SUPERVISED AND UNSUPERVISED SEGMENTATION OF TEXTURED IMAGES BY EFFICIENT MULTI-LEVEL PATTERN CLASSIFICATION
Jaime Christian Meléndez Rodríguez
ISBN:978-84-693-7671-3/DL:T-1750-2010

to k-means, the mode finding process is not controlled by a predefined number of clusters, but by a resolution parameter (bandwidth). Since no assumption regarding the underlaying density function is made, mean shift is said to be able to deal with arbitrarily shaped clusters. However, it can not be directly applied to the present problem, as, in many cases, informative parts of an image seldom coincide with well-separated density modes (Jurie and Triggs, 2005). Furthermore, in the context of image segmentation (e.g., Arbeláez et al., 2009), it has been observed that mean shift tends to produce oversegmentation, which, in our context, would be equivalent to having a high number of prototypes.

Inspired by the last two approaches, a resolution-driven clustering (RDC) algorithm, which is a variation of k-means, is proposed. As with mean shift, the search for clusters is not driven by the number of clusters, but by a resolution parameter R. However, this parameter is defined as a fraction of the longest diagonal of the hypercube that bounds the whole feature space. Thus, it determines the size of the clusters instead of the size of a shift. Moreover, as in g-means, a two-stage approach consisting of split and refinement is followed:

1. *Split*: Suppose that there are already C clusters with their respective centroids modeling the feature space. For each cluster, the hypercube that delimits its volume is found and the length of its longest diagonal is computed. If this value is greater than the resolution parameter R, then that cluster is split into two. Therefore, after a single pass of the algorithm, there will be between C (if no clusters were split) and $2C$ (in case all previous clusters were split) new clusters. The split of a cluster is deterministically done by dividing the bounding box of the original cluster by its longest dimension. New centroids are set at the center of the two resulting bounding boxes.

2. *Refinement*: Once the splitting is complete, the refinement stage follows. It simply consists of the traditional k-means, but using all the resulting centroids

UNIVERSITAT ROVIRA I VIRGILI
SUPERVISED AND UNSUPERVISED SEGMENTATION OF TEXTURED IMAGES BY EFFICIENT MULTI-LEVEL PATTERN CLASSIFICATION
Jaime Christian Meléndez Rodríguez
ISBN:978-84-693-7671-3/DL:T-1750-2010

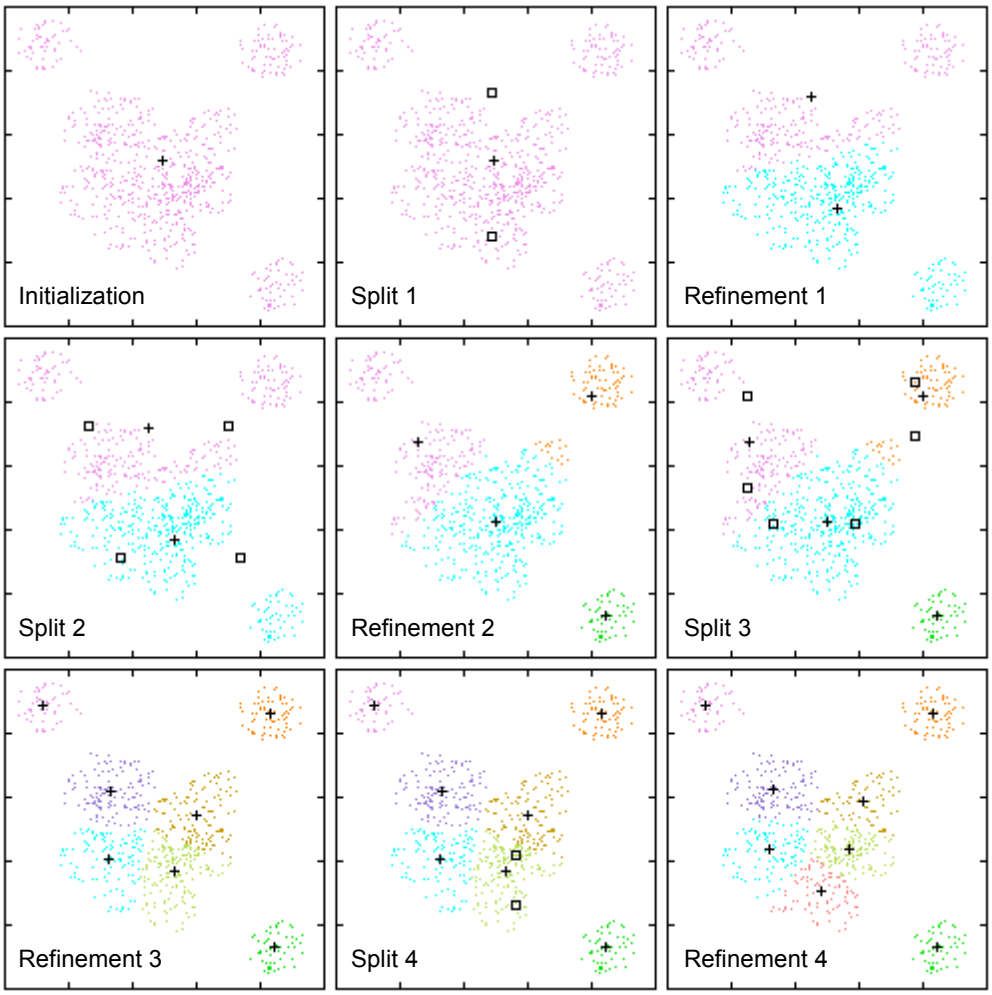

Figure 3.2: A 2-D example of the split and refinement procedure performed by the RDC algorithm. Squares indicate the new initial centroids after split, while crosses indicate the centroids after refinement ($R = 0.3$).

as initial seeds. The k-means algorithm iterates until convergence.

Both the split and refinement stages are repeated until all clusters meet the resolution criterion R. The algorithm is always initialized with a single cluster, whose centroid is the average of all the available feature vectors. Figure 3.2 shows a 2-D example of the clustering process described above.

The proposed RDC algorithm has at least three advantages over the traditional

k-means. First, there is no need for setting the desired number of clusters beforehand. In turn, the desired resolution is introduced and the algorithm finds out the number of clusters that satisfy it. However, a resolution parameter is more intuitive than a number of clusters when dealing with image segmentation problems, especially in high-dimensional spaces. Second, the algorithm always behaves in the same way given the same input points. Therefore, due to its deterministic nature, there is no need for running different trials and keeping the best set of centroids according to some measure, as it is the case when the initialization stage of k-means has a random component. Third, it can handle a variable number of prototypes per texture class given a fixed resolution, a property that makes it more flexible when modeling a complex feature space.

Compared to mean shift and g-means, the RDC approach is more suitable for the classification scheme developed in this work. In particular, experiments in Section 3.5.5 show that the number of prototypes necessary for the classification task found by the proposed approach is smaller than the one obtained with mean-shift and considerably smaller than the one determined by g-means, since the latter tends to overestimate the number of clusters. The reason is that R does not have to be too small (or alternatively there is no need for having a very large number of prototypes) in order to distinguish among the given textures. In addition, the proposed clustering scheme does not make any assumptions about the data distribution, in contrast to g-means, which looks for Gaussian sets.

On the other hand, the main disadvantage of the RDC algorithm is that, due to the followed split-refinement scheme, the time necessary to converge is larger than that of k-means, especially when the size of the sought clusters is too small (e.g., $R = 0.1$). However, since this procedure is run during the training stage, classification times are not affected. Moreover, as mentioned above, there is no need for R to be too small in order to achieve good classification rates in practice.

UNIVERSITAT ROVIRA I VIRGILI
SUPERVISED AND UNSUPERVISED SEGMENTATION OF TEXTURED IMAGES BY EFFICIENT MULTI-LEVEL PATTERN CLASSIFICATION
Jaime Christian Meléndez Rodríguez
ISBN:978-84-693-7671-3/DL:T-1750-2010

3.3.1.2 Normalized Cut Clustering

A different type of partitioning techniques is related to the graph theoretic formulation of grouping. According to this formulation, the set of points in the feature space is represented as a weighted undirected graph $G = (V, E)$, whose nodes are the points in the feature space and the weight of the edge $\omega(a, b)$ between every pair of nodes (a, b) is a function of the similarity between them. Then, a graph can be partitioned into two disjoint sets $\{A, B\}$ by removing edges connecting the two parts.

The dissimilarity between these two sets, which is called a *cut*, can be computed as the total weight of the edges that have been removed:

$$cut(A, B) = \sum_{u \in A, v \in B} \omega(u, v). \tag{3.1}$$

Thus, the optimal bipartition of a graph is the one that minimizes this cut value.

Wu and Leahy (1993) proposed a clustering method that recursively bisects the existing segments based on this *minimum cut* criterion. However, it favors cutting small sets of isolated nodes in the graph, which results in clusters constituted by a single point. To avoid such a bias, Shi and Malik (2000) proposed the *normalized cut*, which, instead of computing the total edge weight connecting the two partitions, measures the cut cost as a fraction of the total edge connections to all the nodes in the graph:

$$Ncut(A, B) = \frac{cut(A, B)}{assoc(A, V)} + \frac{cut(A, B)}{assoc(B, V)}, \tag{3.2}$$

where $assoc(A, V) = \sum_{u \in A, t \in V} \omega(u, t)$ is the total connection from the nodes in A to all nodes in the graph, and $assoc(B, V)$ is defined in a similar way. A computational technique for minimizing this principle based on a generalized eigenvalue problem and its extension in order to obtain a k-way partition by using the top eigenvectors was also proposed in (Shi and Malik, 2000).

Due to its optimal partition properties, the normalized cut clustering (NCC) algorithm has been successfully applied to image segmentation problems (Malik et al.,

1999; Shi and Malik, 2000) and, in this thesis, it is proposed as a means for determining a set of prototypes for each texture pattern. In this context, each node of the graph is the feature vector computed for every image pixel, as described in Section 3.2, and the similarity (or dissimilarity) measure between every pair of nodes is the Euclidean distance between them in the feature space:

$$d_{ab} = \|\mathbf{f}_a(c_1, c_2) - \mathbf{f}_b(c_1, c_2)\|_2. \tag{3.3}$$

In consequence, the graph edge weight connecting those nodes is

$$\omega_{ab} = \exp(\frac{-d_{ab}^2}{\sigma^2}), \qquad \sigma = 0.05 \max(d_{ab}), \forall a, b. \tag{3.4}$$

3.3.1.3 Automatic Parameter Selection

The classifier described in the above sections has two free parameters to be configured: the number of neighbors K evaluated during classification and the resolution R that limits the clusters' size for the RDC algorithm or, alternatively, the number of clusters C for the NCC algorithm. Since the classification performance greatly depends on the choice of those parameters, a procedure for appropriate parameter selection is proposed in this section. Since both R and C are related to the same issue (the number of prototypes for the KNN classifier), although in different algorithms, explanations are given only for the former and can be directly applied to the latter, unless the contrary is explicitly said.

The proposed parameter selection algorithm aims at determining the most suitable combination of K and R based on its performance in classifying the training samples associated with the classes of interest. Since an exhaustive search for (K, R) pairs is prohibitive, only a number of representative values for K, $K_{min} \leq K \leq K_{max}$, and R, $R_{min} \leq R \leq R_{max}$, are explored. Therefore, considering a set of T texture models of interest $\{\tau_1, ..., \tau_T\}$, with every texture model being represented by a sample image $\mathbf{I}_k$, $1 \leq k \leq T$, and W evaluation windows of different size $\{w_1, ..., w_W\}$, the KNN

classifier configured with a specific pair (K, R) is evaluated at each of the valid pixels contained in every image $\mathbf{I}_k$ by using the feature vectors corresponding to window w_j.

The next step consists of comparing the output of the classifier with the label class associated with each training sample and then calculating the classification rate $CR_{(K,R)jk}$, $CR_{(K,R)jk} \in [0, 100]$, associated with each texture model τ_k. Taking into account all T texture models, an average classification rate for the analyzed (K, R) pair can be obtained by

$$CR_{(K,R)j} = \frac{1}{T} \sum_{k=1}^{T} CR_{(K,R)jk}. \tag{3.5}$$

Finally, the best $(K, R)_j$ pair for the considered evaluation window w_j is the one associated with the maximum average classification rate:

$$(K, R)_j^* = \underset{(K,R)_j}{\operatorname{argmax}} \; CR_{(K,R)j}. \tag{3.6}$$

One thing to note is that all (K, R) pairs are not feasible. For instance, given a certain value of R, let $P_{k,min}$ be the number of prototypes of the texture pattern with the minimum number of clusters according to the prototype extraction algorithm described in Section 3.3.1.1 (for the NCC algorithm in Section 3.3.1.2, all patterns have the same number of clusters given C; therefore, $P_{k,min} = C$). In order to guarantee that all texture patterns can be identified, K must be lower than or equal to $P_{k,min}$, as without this constraint, it is possible to reach a point in the voting phase of the KNN classifier where all the $P_{k,min}$ available votes for that texture pattern have already been used (the available votes for a texture pattern coincide with the number of its modeling prototypes) and, as K is greater than $P_{k,min}$, the remaining $K - P_{k,min}$ votes will only consider the textures that can still vote. Thus, only pairs with $K \leq P_{k,min}$ must be evaluated.

Taking into account the multi-level classification scheme of Section 3.2, the selection procedure should be repeated W times in order to select the most appropriate parameters for every evaluation window size. However, by specifying a threshold ζ,

UNIVERSITAT ROVIRA I VIRGILI
SUPERVISED AND UNSUPERVISED SEGMENTATION OF TEXTURED IMAGES BY EFFICIENT MULTI-LEVEL PATTERN CLASSIFICATION
Jaime Christian Meléndez Rodríguez
ISBN:978-84-693-7671-3/DL:T-1750-2010

defined as the *minimum, admissible classification rate* considering the training set, a new number of windows to be evaluated W^*, $W^* \leq W$, can be determined. The idea is to stop the search for K and R if the classification rate surpasses this minimum, admissible threshold and to make the current evaluation window size the new maximum $s_{W^*} \times s_{W^*}$, instead of the previous $s_W \times s_W$.

By setting a suitable value for ζ (e.g., $\zeta = 99$), a considerable improvement in classification time can be achieved without compromising accuracy, by discarding the largest window sizes, which are the most expensive to evaluate and which are not always necessary for good discrimination among texture patterns. Furthermore, recalling that in texture segmentation, small but competitive evaluation windows are desired in order to precisely locate the boundaries between regions, the aforementioned threshold significantly contributes towards that goal.

The proposed parameter selection algorithm is evaluated in Section 3.5.3.

3.3.2 Support Vector Machine-Based Classifier

Another way to efficiently summarize and learn all the available information obtained from the training set is through SVMs. In the context of classification, the role of support vectors (SVs) can be thought of as complementary to the role of prototypes in the following sense: while prototypes model the interior of the relevant portions of the clouds of feature points associated with the analyzed patterns, SVs model the frontiers of those clouds.

An SVM classifier maps an M-dimensional data point $\mathbf{x}$ into a class label y, $y \in \{\pm 1\}$, based on an aggregating decision function of the form

$$D(\mathbf{x}) = \mathbf{w}^\top \phi(\mathbf{x}) + \beta, \tag{3.7}$$

where ϕ is a function that maps $\mathbf{x}$ form its original M-dimensional space into a (new) L-dimensional space, $\mathbf{w}$ is a weight vector and β is a bias term.

UNIVERSITAT ROVIRA I VIRGILI
SUPERVISED AND UNSUPERVISED SEGMENTATION OF TEXTURED IMAGES BY EFFICIENT MULTI-LEVEL PATTERN CLASSIFICATION
Jaime Christian Meléndez Rodríguez
ISBN:978-84-693-7671-3/DL:T-1750-2010

Since an SVM is a binary classifier, an extension is needed in order to solve multiclass problems. Several methods that construct a multiclass classifier by combining a number of binary classifiers have been proposed: *one-against-all* SVMs (Vapnik, 1998), where one class is separated from the remaining classes; *one-against-one* SVMs (Kreßel, 1999), where every single class is separated from any other class; and *directed acyclic graph* SVMs (Platt. et al., 2000), whose training phase is the same as the one-against-one method, but whose testing phase uses a rooted binary acyclic graph where each node is a binary SVM. In addition, methods that consider all classes at once, i.e., which compute decision functions by solving only one, but larger problem, also exist (Vapnik, 1998; Crammer and Singer, 2002).

Comparative results in (Tsujinishi et al., 2004) suggest that one-against-one SVMs are the best alternative in terms of both classification accuracy and computation time, as only "small", "easy" two-class problems need to be solved, which, in addition, yields a reduced number of SVs. These results have also been corroborated in our context through preliminary experimentation. Therefore, the one-against-one extension has been chosen for supervised pixel-based classification.

Under that extension, a binary classifier is required for each pair of classes. Thus, given a classification problem with T classes, the total number of SVMs is $T(T-1)/2$. The final classification considering every pair of classes (k, k') is obtained by the following rule (Hsu and Lin, 2002): if according to $\text{sign}(D(\mathbf{x}))$, $\mathbf{x}$ belongs to class k, then the votes for class k are incremented by one. Otherwise, the votes associated with class k' are incremented by one. Finally, $\mathbf{x}$ is assigned to the class with the largest number of votes.

Since only the sign of the decision function is considered, the above strategy is discrete, that is, each binary classifier directly chooses the best alternative and the aggregated score only accounts for that alternative, completely disregarding the other. However, this may lead to wrong decisions in cases where both alternatives are almost equally likely (e.g., when $\mathbf{x}$ is close to the separating hyperplane). Furthermore, two

UNIVERSITAT ROVIRA I VIRGILI
SUPERVISED AND UNSUPERVISED SEGMENTATION OF TEXTURED IMAGES BY EFFICIENT MULTI-LEVEL PATTERN CLASSIFICATION
Jaime Christian Meléndez Rodríguez
ISBN:978-84-693-7671-3/DL:T-1750-2010

or more classes may have identical votes and, in consequence, an additional untying mechanism would be necessary. Therefore, instead of discrete votes, probability estimates have been included. In this way, for the final classification, probabilities for each class are added and $\mathbf{x}$ is assigned to the class with the highest probability. Probability estimates for SVMs have been computed as in (Platt, 2000) by approximating the posterior by a sigmoid function:

$$Pr(k|k \text{ or } k', \mathbf{x}) = \frac{1}{1 + \exp(\gamma_1 D(\mathbf{x}_n) + \gamma_2)}, \tag{3.8}$$

where $D(\mathbf{x}_n)$, $n \in [1, N]$, are the decision values of training data, and γ_1 and γ_2 are estimated by solving a regularized maximum likelihood problem as described in (Lin et al., 2007).

Since SVs and decision functions are determined by the method utilized for SVM optimization, there is no need for a complete parameter selection algorithm as with the previous classifier. In fact, only the last part, which aims at reducing the number of evaluation windows given the minimum, admissible classification rate ζ is taken into account.

3.4 Efficiency Improvement

The methodology for pixel-based classification proposed in the previous sections is already efficient in the sense that it effectively summarizes the information supplied in the training stage, either by determining suitable prototypes or support vectors, and it uses this information in a clever way by following the proposed multi-level classification approach. Notwithstanding, further efficiency improvements can be achieved by following the three concrete strategies discussed in this section, namely: reducing the number of prototypes for the KNN classifier, reducing the number of SVs for the SVM-based classifier and performing feature selection for both classifiers.

3.4.1 Reducing the Number of Prototypes

One of the main factors that influences the classification time of the KNN classifier in Section 3.3.1 is the number of prototypes obtained during training. With respect to this factor, the method for selecting an appropriate R or C described in Section 3.3.1.3 aims at determining the best set of prototypes by only considering the highest classification rate, but disregarding the number of prototypes with which it is achieved. Therefore, in terms of efficiency, it can be further improved.

An alternative is to include some heuristic with the goal of maximizing the classification rate and, at the same time, minimizing the number of resulting prototypes, thus establishing a better tradeoff between *benefit* and *cost*. Such a heuristic has been given in (Puig and Garcia, 2006), although in the context of feature selection. It defines a performance measure that takes into account the benefit of integrating a number of features and the cost associated with those features through the following quotient:

$$\rho = \frac{1}{3}\left(2\frac{\text{Benefit}+1}{\text{Cost}+1} - 1\right), \tag{3.9}$$

were both Benefit and Cost are sets of values normalized in the $[0,1]$ interval in order to be comparable. ρ gets its maximum value ($\rho = 1$) in case the maximum benefit (benefit $= 1$) is obtained with the minimum cost (cost $= 0$), and its minimum value ($\rho = 0$) if the minimum benefit (benefit $= 0$) is obtained with the maximum cost (cost $= 1$). In the end, the number of most significant features is selected by

$$\eta = \underset{\text{cost}}{\text{argmax}}\ \rho\text{Benefit}. \tag{3.10}$$

One thing to note is that the final selection could have directly been carried out by

$$\eta = \underset{\text{cost}}{\text{argmax}}\ \rho, \tag{3.11}$$

instead of by equation (3.10). However, no reasons are given in (Puig and Garcia, 2006) for choosing the latter. Moreover, by incorporating Benefit, equation (3.10) is

UNIVERSITAT ROVIRA I VIRGILI
SUPERVISED AND UNSUPERVISED SEGMENTATION OF TEXTURED IMAGES BY EFFICIENT MULTI-LEVEL PATTERN CLASSIFICATION
Jaime Christian Meléndez Rodríguez
ISBN:978-84-693-7671-3/DL:T-1750-2010

somewhat biased to maximizing the benefit to the detriment of minimizing the cost. In fact, consider the example shown in Figure 3.3(a), where the green line corresponds to cost versus benefit pairs, the blue line corresponds to ρ associated with those pairs and the red line corresponds to ρBenefit also associated with those pairs. Maximums of ρ and ρBenefit are marked with squares of their respective colors over the original cost versus benefit curve. As it can be appreciated, the selection using equation (3.11) has a lower cost than the selection through equation (3.10).

Now, consider the example shown in Figure 3.3(b). Relevant curves and points are indicated in the same way as in Figure 3.3(a). Again, the selection by equation (3.11) has a lower cost than the selection by equation (3.10). However, the difference in benefit between the two solutions is maximum in contrast to the previous example, where it was very small. Thus, in this second case, the selection by equation (3.10) is preferable, as it avoids a considerable degradation in benefit and, at least, guarantees the same performance as the original algorithm (i.e., without applying any heuristic to reduce its cost).

As a consequence, the key point is how to distinguish one situation from the other in order to apply equation (3.10) or equation (3.11). To solve this problem, a *performance limit* is defined based on the following reasoning: since both cost and benefit are comparable (both are normalized in the $[0, 1]$ interval), an efficient process is the one that produces a benefit superior to its cost. Then, considering cost versus benefit pairs where benefit = cost, a *performance limit curve* $\widehat{\rho}$ can be computed. It corresponds to the black line shown in Figures 3.3(a) and 3.3(b). Taking into account this new definition, the final selection is now performed by

$$\eta = \begin{cases} \underset{\text{cost}}{\text{argmax}}\ \rho & \text{if } \rho > \widehat{\rho} \\ \underset{\text{cost}}{\text{argmax}}\ \rho\text{Benefit} & \text{otherwise.} \end{cases} \tag{3.12}$$

In practice, $\rho > \widehat{\rho}$ if most of the values in the ρ set are greater than the values in the $\widehat{\rho}$ set, except for the first and the last values if they coincide with the minimum and

UNIVERSITAT ROVIRA I VIRGILI
SUPERVISED AND UNSUPERVISED SEGMENTATION OF TEXTURED IMAGES BY EFFICIENT MULTI-LEVEL PATTERN CLASSIFICATION
Jaime Christian Meléndez Rodríguez
ISBN:978-84-693-7671-3/DL:T-1750-2010

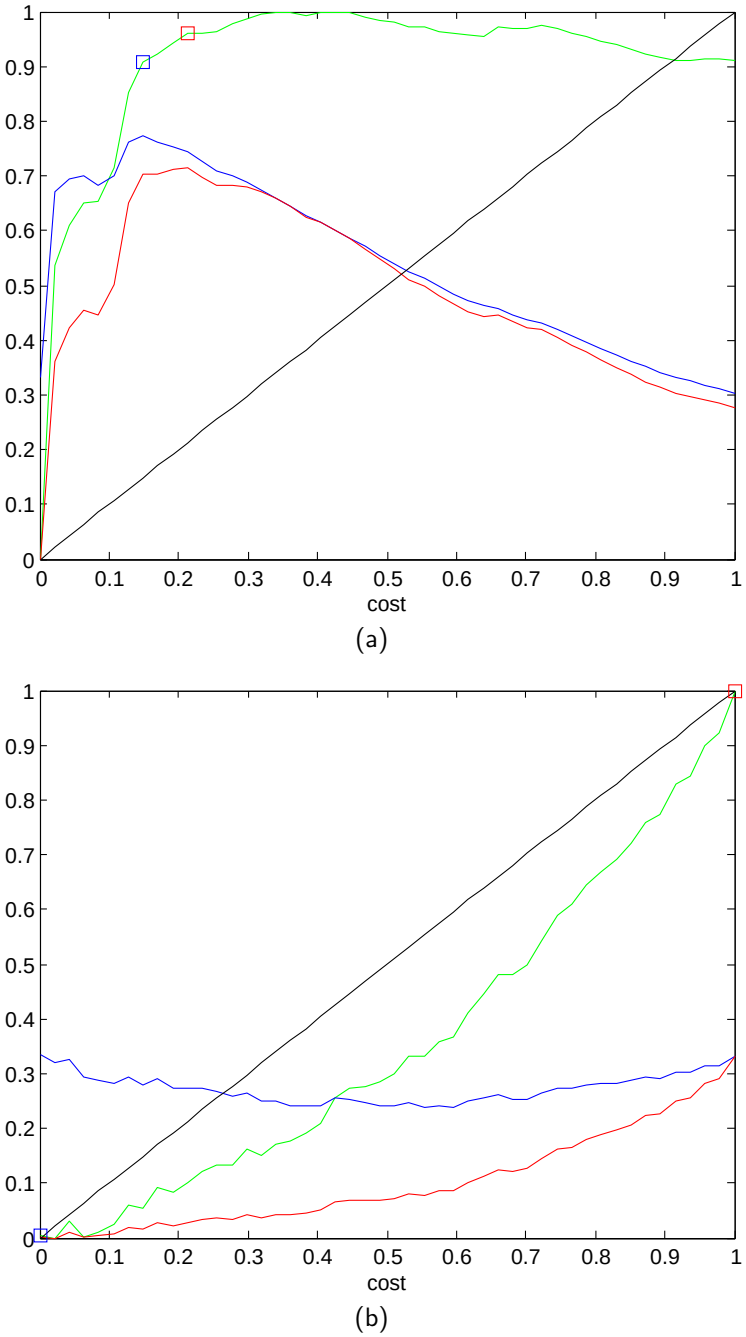

Figure 3.3: Examples of cost versus benefit problems found in this thesis.

maximum, respectively.

In order to apply equations (3.9) and (3.12) to the current problem, Benefit is

UNIVERSITAT ROVIRA I VIRGILI
SUPERVISED AND UNSUPERVISED SEGMENTATION OF TEXTURED IMAGES BY EFFICIENT MULTI-LEVEL PATTERN CLASSIFICATION
Jaime Christian Meléndez Rodríguez
ISBN:978-84-693-7671-3/DL:T-1750-2010

estimated by the classification rates computed according to equation (3.5) and Cost is estimated by the total number of prototypes P associated with each classification rate. We use $P_{(R)}$ or $P_{(C)}$ depending on whether the RDC or the NCC algorithm has determined those prototypes. Thus, by considering the RDC algorithm again (the formulation for the NCC algorithm is equivalent), the parameter selection algorithm in Section 3.3.1.3 can be modified as follows.

First, instead of selecting the best K and R by evaluating equation (3.6), $\rho_{(K)j}$ is computed for each alternative K given evaluation window w_j:

$$\rho_{(K)j} = \frac{1}{3}\left(2\frac{\widetilde{CR}_{(R)j}+1}{\widetilde{P}_{(R)j}+1} - 1\right), \tag{3.13}$$

where both $\widetilde{CR}_{(R)j}$ and $\widetilde{P}_{(R)j}$ are values normalized between zero and one: $\widetilde{CR}_{(R)j} = (CR_{(R)j} - CR_{(R)j,min})/(CR_{(R)j,max} - CR_{(R)j,min})$, $\widetilde{P}_{(R)j} = (P_{(R)j} - P_{(R)j,min})/(P_{(R)j,max} - P_{(R)j,min})$, with $CR_{(R)j,min}$ and $CR_{(R)j,max}$ being the minimum and maximum average classification rates, and $P_{(R)j,min}$ and $P_{(R)j,max}$ being the minimum and maximum number of evaluated prototypes, respectively. Then, $R^*_{(K)j}$ associated with the best number of prototypes for each alternative K is determined by

$$R^*_{(K)j} = \begin{cases} \underset{P_{(R)j}}{\text{argmax}}\ \rho_{(K)j} & \text{if } \rho_{(K)j} > \widehat{\rho}_{(K)j} \\ \underset{P_{(R)j}}{\text{argmax}}\ \rho_{(K)j}\widetilde{CR}_{(R)j} & \text{otherwise,} \end{cases} \tag{3.14}$$

where $\widehat{\rho}_{(K)j}$ is made up of pairs that satisfy $\widetilde{CR}_{(R)j} = \widetilde{P}_{(R)j}$. Finally, $(K,R)^*_j$ is determined as the best pair associated with the maximum average classification rate:

$$(K,R)^*_j = \underset{K_j, R^*_{(K)j}}{\text{argmax}}\ CR_{(K_j, R^*_{(K)j})}. \tag{3.15}$$

Since no substantial degradation in the classification rate is expected, the remaining part of the parameter selection algorithm concerning the reduction of the number of evaluation windows is the same. Results of applying this prototype reduction methodology are discussed in Section 3.5.7 (First Part).

3.4.2 Reducing the Number of Support Vectors

Although the one-against-one strategy adopted for the SVM classifier in Section 3.3.2 is the one that usually yields the smallest number of SVs compared to the remaining alternative multiclass extensions, hundreds or even thousands of these elements may still need to be evaluated depending on the separability of the modeled texture classes. Thus, by reducing this number, a considerable speedup in classification can be achieved.

According to Tsujinishi et al. (2004), that kind of simplification is feasible if a $\mathbf{z}$ satisfying $\phi(\mathbf{z}) = \mathbf{w}$ exists, since it would be possible to evaluate the single decision function $D(\mathbf{x}) = \Phi(\mathbf{x}, \mathbf{z}) + \beta$, instead of evaluating a separate decision function for each SV. In practical terms, it would be the same as evaluating a single SV.

In the above formulation, $\Phi(\mathbf{x}, \mathbf{x}') = g(\mathbf{x}^\top \mathbf{x}')$, where $g(\cdot)$ is some scalar function. Then,

$$\Phi(\mathbf{z}, \mathbf{e}_i) = \mathbf{w}^\top \phi(\mathbf{e}_i) = g(z^i) = \sum_{n=1}^{N} \alpha_n y_n x_n^i, \tag{3.16}$$

where $\mathbf{e}_i$ is a basis vector in the input space with its ith element being one and the others zero, α_n are the coefficients computed during optimization of the SVM, and x_n^i is the ith element of training data point $\mathbf{x}_n$. As a consequence, the ith component of $\mathbf{z}$ can be computed as

$$z^i = g^{-1}\left(\sum_{n=1}^{N} \alpha_n y_n x_n^i\right). \tag{3.17}$$

Although this equation considers every training point $\mathbf{x}_n$, as it was originally proposed in the context of least-square SVMs, equivalent results are achieved by only considering SVs instead of every training point.

For linear kernels, $\phi^{-1}(\mathbf{w}) = \mathbf{w}$. Thus, by performing the sum in equation (3.17), all the information modeled through the SVs can be fused into a single element. However, if an M-dimensional vector $\mathbf{x}$ is mapped to an L-dimensional space ($L > M$), the inverse of $\phi(\mathbf{z})$ does not exist, as a set of L equations must be satisfied for

UNIVERSITAT ROVIRA I VIRGILI
SUPERVISED AND UNSUPERVISED SEGMENTATION OF TEXTURED IMAGES BY EFFICIENT MULTI-LEVEL PATTERN CLASSIFICATION
Jaime Christian Meléndez Rodríguez
ISBN:978-84-693-7671-3/DL:T-1750-2010

M variables (Tsujinishi et al., 2004). For instance, considering a polynomial kernel of degree 2 and a one-dimensional input x, $\Phi(x, x') = (1 + xx')^2$. Therefore, $\phi(x)$ is given by $\phi(x) = (1, \sqrt{2}x, x^2)^\top$ and the following system of equations must be satisfied:

$$1 = r_1 + r_2, \tag{3.18}$$

$$z = r_1 x_1 + r_2 x_2, \tag{3.19}$$

$$z^2 = r_1 x_1^2 + r_2 x_2^2, \tag{3.20}$$

which is, in general, unsolvable. Thus, the above reduction of SVs is only applicable to SVMs with linear kernels. The performance of linear kernels for pixel-based texture classification is evaluated in Section 3.5.6, while the speedup in classification due to the SV reduction method is demonstrated in Section 3.5.7 (First Part).

3.4.3 Feature Selection

By applying a similar strategy as in Section 3.4.1, although this time measuring the cost in terms of texture features instead of texture prototypes, the number of features processed by either the KNN classifier or the SVM-based classifier can be reduced. In this case, classification rates are used for assessing the *significance* of each feature, as well as for representing the benefit of processing a given subset of features. In order to build analogous cost versus benefit curves, a sequential forward generation procedure considering a sorted list of features is utilized. The selection procedure is repeated W^* times.

The first step of the feature selection algorithm consists of determining the significance of each feature. Thus, given a set of M features $\{f_1, ..., f_M\}$, a set of T texture models of interest $\{\tau_1, ..., \tau_T\}$ and W^* evaluation windows of different size $\{w_1, ..., w_{W^*}\}$, one of the previously configured classifiers is run in order to classify the training data samples into one of the T known classes, only considering feature f_i, $1 \leq i \leq M$, and window w_j, $1 \leq j \leq W^*$. Then, the classification rate CR_{ijk}, $CR_{ijk} \in [0, 100]$, associated with each class τ_k is computed and the *individual*

significance S_{ijk} of feature f_i with respect to class τ_k is defined as the normalized classification rate

$$S_{ijk} = CR_{ijk} / \sum_{i=1}^{M} CR_{ijk}. \tag{3.21}$$

S_{ijk} is defined in the interval $[0, 1]$, with zero indicating that class τ_k cannot be distinguished by using feature f_i, and one, that f_i is the only feature through which τ_k can be classified. Now, the *global significance* of feature f_i considering all the classes of interest is computed by

$$S_{ij} = S_{ijk} / \sum_{k=1}^{T} S_{ijk}. \tag{3.22}$$

Taking equation 3.22 into account, a ranking with all the available features is defined. The feature at the top of the ranking is the one with the highest global significance, while the last feature in the list is the one with the lowest. Thus, a list sorted in descending order is created.

The next step consists of choosing a specific subset of features out of that sorted list. In order to do this, the training samples are subsequently classified by processing different subsets of features, starting with the most significant feature and progressively adding a new feature from the head of the sorted list until all features are included. Average classification rates CR_{mj} for each subset of m features, $1 \leq m \leq M$, considering the T classes of interest are computed by

$$CR_{mj} = \frac{1}{T} / \sum_{k=1}^{T} CR_{mjk}, \tag{3.23}$$

where CR_{mjk} are the classification rates obtained for each individual class τ_k when the first m features are processed.

Then, both CR_{mj} and m are normalized between zero and one: $\widetilde{CR}_{mj} = (CR_{mj} - CR_{mj,min})/(CR_{mj,max} - CR_{mj,min})$, $\widetilde{m} = (m-1)/(M-1)$, with $CR_{mj,min}$ and $CR_{mj,max}$ being the minimum and maximum average classification rates, and the performance measure is defined as

$$\rho_j = \frac{1}{3}\left(2\frac{\widetilde{CR}_{mj} + 1}{\widetilde{m} + 1} - 1\right). \tag{3.24}$$

UNIVERSITAT ROVIRA I VIRGILI
SUPERVISED AND UNSUPERVISED SEGMENTATION OF TEXTURED IMAGES BY EFFICIENT MULTI-LEVEL PATTERN CLASSIFICATION
Jaime Christian Meléndez Rodríguez
ISBN:978-84-693-7671-3/DL:T-1750-2010

Finally, the M_j^* most significant features for evaluation window w_j are selected by

$$M_j^* = \begin{cases} \underset{m}{\text{argmax}}\ \rho_j & \text{if } \rho_j > \widehat{\rho}_j \\ \underset{m}{\text{argmax}}\ \rho_j \widetilde{CR}_{mj} & \text{otherwise,} \end{cases} \tag{3.25}$$

where $\widehat{\rho}_j$ is the performance limit curve as defined in Section 3.4.1, which, this time, is made up of pairs that satisfy $\widetilde{CR}_{mj} = \widetilde{m}$.

Experimental results corresponding to this feature selection algorithm are shown in Section 3.5.7 (Second Part). Results of jointly applying the strategies proposed in this section are shown in Section 3.5.7 (Third Part).

3.5 Experimentation

This section illustrates the application of the proposed pixel-based classifier to supervised image segmentation. It describes the experimental setup and how the pixel-based classifier is configured for the performed experiments, and also shows and discusses the obtained results. Five sets of experiments are carried out. In the first set of experiments, the effectiveness of the automatic parameter selection algorithm proposed in Section 3.3.1.3 is assessed. In the next set of experiments, the advantages of incorporating the multi-level classification scheme of Section 3.2 are demonstrated. In the third set of experiments, alternatives for prototype computation for the KNN classifier are compared to the methods proposed in Sections 3.3.1.1 and 3.3.1.2. In the fourth set of experiments, the performance of different, potential kernel functions for the SVM-based classifier is considered. Finally, in the last set of experiments, efficiency improvements due to the strategies in Section 3.4 are evaluated.

3.5.1 Experimental Setup

The proposed technique has been evaluated on a wide variety of synthetic and real images corresponding to sets a to d shown in Appendix A (Figures A.1, A.2, A.4

and A.6). It is important to highlight that, although the boundary locations are exactly known for the synthetic compositions (sets a and b), only approximate, subjective boundaries can be associated with the real scenes (sets c and d). Therefore, segmentation quality results should be considered differently for both types of images.

The experimental methodology follows the one described in (Randen and Husøy, 1999). In this sense, each test image represents a separate problem and has its own training patterns, although, in practice, some or all patterns may also correspond to different test images. For the real images, both test and training samples belong to non-overlapping portions of the same scene and have been acquired under the same lighting conditions. Other factors such as scale and perspective may (intentionally) change in some cases. In addition, there are instances of the real images where not all training patterns appear on the test image (images 17, 20, 21, 24, 25). Figure 3.4 shows examples of both synthetic and real test images and their associated training patterns.

The segmentation quality is measured in terms of classification rate, which is defined as the ratio between the number of correctly classified pixels and the number of valid pixels in the ground-truth. This score is bounded in the $[0, 100]$ interval. While for the synthetic compositions, all the pixels in the test image (as well as in the ground-truth) are considered to be valid, as their patterns are accurately known to belong to a given class, for the real scenes, some patterns in the test image may not match any of those learnt during training or may match more than one. Thus, they can not be undoubtedly assigned to one of the sought patterns and are said to belong to "unknown" classes. As a result, they are not taken into account for computing the classification rate. Along the experiments, these unknown classes are represented as black regions in the corresponding ground-truth as in the examples shown in Figure 3.5. In addition, the strip of unclassified pixels at the boundary of the processed image due to the applied evaluation window size (or the smallest evaluation window size in the case of a multi-window approach), as discussed in

UNIVERSITAT ROVIRA I VIRGILI
SUPERVISED AND UNSUPERVISED SEGMENTATION OF TEXTURED IMAGES BY EFFICIENT MULTI-LEVEL PATTERN CLASSIFICATION
Jaime Christian Meléndez Rodríguez
ISBN:978-84-693-7671-3/DL:T-1750-2010

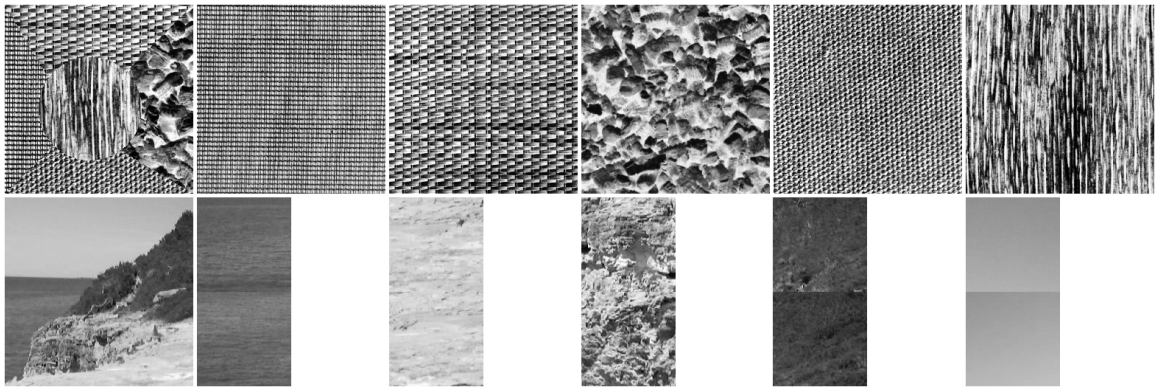

Figure 3.4: Examples of test images and their associated training patterns. Synthetic composition (top). Real outdoor scene (bottom).

Figure 3.5: Examples of test images with "unknown" patterns and their corresponding ground-truths.

Section 3.2, is not considered valid either. This strip of pixels is shown in black in the corresponding classification maps.

Computation times for all experiments in this chapter correspond to an Intel Pentium 4 at 3.2 GHz with 2 GB of RAM. They are always given in seconds.

3.5.2 Classifier Configuration

The configuration of the various stages involved in the proposed pixel-based texture classification scheme has been carried out both manually, by taking into account previous related works, or automatically, by the selection algorithms described in the previous sections. Details about this issue are given below.

During the *feature extraction stage*, images are filtered with a multichannel Gabor

filter bank by following the design strategy proposed in (Manjunath and Ma, 1996). This strategy has widely been utilized for texture classification and segmentation tasks (e.g., Manjunath and Ma, 1996; Ma and Manjunath, 2000; Muneeswaran et al., 2006), and has even been adopted as texture descriptor by the MPEG-7 standard (Manjunath et al., 2001). Moreover, it has shown superior results in the context of pixel-based classification when compared to other relevant texture feature extraction alternatives (Melendez et al., 2007).

The Gabor filter bank has been configured with four scales and six orientations, and a range of frequencies between 0.05 and 0.4 according to (Manjunath and Ma, 1996). After filtering an input image, the texture features that characterize every pixel and its surrounding neighborhood are the mean and standard deviation of the module of the resulting values. Therefore, every feature vector has a total of $M = 6 \times 4 \times 2 = 48$ dimensions. All dimensions are normalized between 0 and 1, thus avoiding any bias.

In order to adapt the above texture feature extraction procedure to the multi-level classification scheme proposed in Section 3.2, features have been computed for W different evaluation window sizes as suggested in (Puig and Garcia, 2006; Melendez et al., 2008), although W has been set to 5 in this case, which corresponds to sizes: 3×3, 5×5, 9×9, 17×17 and 33×33. Following the same reasoning, the size of the filter's sampling grid is also variable and is set to be the same as the size of its corresponding evaluation window.

During the *supervised training stage*, different procedures are followed depending on the chosen classification technique. For the KNN classifier, a set of prototypes is determined from the feature vectors extracted from the training images, either by the RDC or the NCC algorithms. As few as 1/16 of the available feature vectors is utilized. This contributes to speeding the process up without compromising the quality of the outcome. Then, in order to determine appropriate parameters for the classifier, the complete parameter selection algorithm described in Section 3.3.1.3 is

UNIVERSITAT ROVIRA I VIRGILI
SUPERVISED AND UNSUPERVISED SEGMENTATION OF TEXTURED IMAGES BY EFFICIENT MULTI-LEVEL PATTERN CLASSIFICATION
Jaime Christian Meléndez Rodríguez
ISBN:978-84-693-7671-3/DL:T-1750-2010

applied. Again, in order to accelerate the process, only 1/4 of the training vectors are used. The values explored by the parameter selection algorithm are: $K = \{1, 2, ..., 5\}$, $R = \{0.1, 0.125, ..., 1.0\}$ for RDC or $C = \{1, 2, ..., 50\}$ for NCC, and $\zeta = 99$. A reduced number of prototypes can be obtained by including the modifications developed in Section 3.4.1.

On the other hand, for the SVM-based classifier, linear kernels are configured in order to take advantage of the SV reduction method given in Section 3.4.2. As in the previous case, 1/16 of the available feature vectors is utilized for training and ζ is set to 99. As an extra step, both the KNN and SVM-based classifiers can benefit from the feature selection algorithm of Section 3.4.3.

During the *classification stage*, a given test image is processed in order to identify the texture pattern corresponding to each of its pixels. Feature extraction, as explained above, is performed in order to obtain a feature vector for every pixel. Then, each vector is classified into one of the given texture patterns by any of the previously configured classifiers following the multi-level approach described in Section 3.2. After pixel-based classification, a post-processing procedure is applied. It consists of two steps. In the first step, a small window (e.g., 5×5 pixels) is displaced pixel by pixel along the entire image and the central pixel of the enclosed region is replaced by its dominant texture class. In the second step, small or thin regions are assigned to the adjacent region with the largest number of neighboring pixels. The size of regions is determined by counting their pixels, while thickness is captured by means of the distance transform.

3.5.3 First Set of Experiments

The goal of the first set of experiments is to validate the methodology for automatic parameter selection proposed in Section 3.3.1.3 (with no modifications for efficiency improvement). Therefore, a similar search has been performed in order to find out the combination of parameters that yields the best classification results, but, this time,

considering the test images instead of the training images. Then, the classification rates produced by the best combinations of parameters have been compared with the classification rates obtained with the parameters automatically chosen by the parameter selection algorithm.

The techniques evaluated in these experiments correspond to those described in Section 3.3. The KNN classifier fed with prototypes determined by the RDC algorithm (Section 3.3.1.1) will be referred to as GMM-RDC-KNN, where GMM represents the Gabor features according to (Manjunath and Ma, 1996). In a similar way, the KNN classifier fed with prototypes determined by the NCC algorithm (Section 3.3.1.2) will be referred to as GMM-NCC-KNN. Finally, the SVM-based classifier with the one-against-one extension (Section 3.3.2) will be referred to as GMM-SVM-OAO.

Tables 3.1 to 3.3 summarize the results of these experiments. They indicate that the proposed selection algorithm is effective in finding reasonable combinations of parameters for every classifier, as the classification rates obtained in this way are very close (or even the same in some cases) to the classification rates obtained with "optimal" parameters. On the other hand, the results also indicate that the classifiers with "optimal" parameters achieve higher classification rates with a lower number of prototypes or SVs, which is more noticeable in the case of GMM-RDC-KNN, where the difference is around 14%. However, this issue can be solved by applying the strategies for reducing the number of prototypes and SVs of Sections 3.4.1 and 3.4.2, as shown in Section 3.5.7. Finally, the number of neighbors and evaluation windows determined by the proposed algorithm is slightly lower, which will contribute to speeding up the classification and feature extraction processes.

One point to remark is that the number of nearest neighbors per window size is, in most cases, set to one (this is appreciated when values for K^* and W^* coincide), considering both the training and test sets for the parameter search, which is quite surprising, since, in theory, more than one neighbor is usually necessary for

UNIVERSITAT ROVIRA I VIRGILI
SUPERVISED AND UNSUPERVISED SEGMENTATION OF TEXTURED IMAGES BY EFFICIENT MULTI-LEVEL PATTERN CLASSIFICATION
Jaime Christian Meléndez Rodríguez
ISBN:978-84-693-7671-3/DL:T-1750-2010

Table 3.1: Classification rates (CR) for GMM-RDC-KNN configured both with "optimal" parameters and by the parameter selection algorithm described in Section 3.3.1.3. The considered parameters are: the total number of prototypes (P^*), the total number of neighbors (K^*) and the number of evaluation windows (W^*).

Image	Classifier with "optimal" parameters				Classifier with selected parameters			
	P^*	K^*	W^*	CR	P^*	K^*	W^*	CR
1	657	3	3	97.51	657	3	3	97.51
2	1039	4	4	95.73	1039	4	4	95.73
3	570	3	3	98.08	570	3	3	98.08
4	889	10	5	92.21	1118	6	4	91.68
5	389	6	4	87.61	1537	7	5	86.22
6	624	4	4	93.16	863	4	4	92.31
7	757	3	3	91.63	746	6	4	91.60
8	879	5	5	88.10	1083	5	5	87.48
9	1395	5	5	94.15	1395	5	5	94.15
10	1207	5	5	90.93	885	4	4	89.82
11	1162	5	5	91.00	1162	5	5	91.00
12	1259	5	5	81.13	1259	5	5	81.13
13	1125	4	4	96.50	1039	4	4	96.23
14	1267	5	5	76.22	1267	5	5	76.22
15	58	3	3	93.11	86	3	3	92.46
16	59	3	3	90.23	86	3	3	89.99
17	55	3	3	91.80	86	3	3	91.40
18	57	3	3	90.22	86	3	3	88.87
19	100	3	3	93.00	280	4	4	92.46
20	191	5	5	91.27	280	4	4	91.24
21	218	5	3	97.83	280	4	4	97.63
22	159	3	3	93.92	256	4	4	92.99
23	128	9	5	91.32	256	4	4	90.63
24	112	8	4	99.33	194	3	3	98.72
25	96	5	3	98.05	194	3	3	97.24
26	138	6	3	97.62	138	4	4	97.03
27	124	5	3	85.74	173	3	3	84.66
28	149	3	3	95.55	266	3	3	94.97
29	152	3	3	89.53	184	3	3	89.50
30	264	3	3	91.41	264	3	3	91.41
31	156	9	5	95.05	178	3	3	93.89
Average	497.90	4.71	3.81	92.22	577.65	3.97	3.77	91.75

UNIVERSITAT ROVIRA I VIRGILI
SUPERVISED AND UNSUPERVISED SEGMENTATION OF TEXTURED IMAGES BY EFFICIENT MULTI-LEVEL PATTERN CLASSIFICATION
Jaime Christian Meléndez Rodríguez
ISBN:978-84-693-7671-3/DL:T-1750-2010

Table 3.2: Classification rates (CR) for GMM-NCC-KNN configured both with "optimal" parameters and by the parameter selection algorithm described in Section 3.3.1.3. The considered parameters are: the total number of prototypes (P^*), the total number of neighbors (K^*) and the number of evaluation windows (W^*).

Image	Classifier with "optimal" parameters				Classifier with selected parameters			
	P^*	K^*	W^*	CR	P^*	K^*	W^*	CR
1	415	4	4	97.44	863	4	4	96.95
2	1057	5	5	95.42	865	4	4	95.07
3	638	3	3	98.21	615	5	3	98.19
4	882	7	5	94.61	882	7	5	94.61
5	1093	9	5	88.21	1093	9	5	88.21
6	925	11	5	93.42	621	4	4	93.20
7	406	4	4	91.11	796	8	5	91.04
8	1016	5	5	91.68	1016	5	5	91.68
9	752	5	5	94.09	738	5	5	94.05
10	704	5	5	90.32	943	5	5	89.87
11	860	5	5	91.79	1107	5	5	91.32
12	660	4	4	78.08	1039	5	5	77.92
13	679	4	4	96.11	810	4	4	96.01
14	621	5	5	74.74	690	5	5	74.68
15	171	3	3	91.01	137	3	3	90.79
16	256	3	3	91.90	137	3	3	91.85
17	285	5	3	92.30	137	3	3	92.28
18	247	7	4	90.14	137	3	3	89.80
19	362	4	4	94.68	282	4	4	94.06
20	334	4	4	93.10	282	4	4	92.98
21	341	4	4	96.89	282	4	4	96.89
22	291	4	4	94.12	379	4	4	93.89
23	397	8	4	91.07	379	4	4	90.65
24	406	5	3	99.11	392	4	4	98.97
25	656	5	5	97.95	392	4	4	97.59
26	516	8	4	94.66	278	3	3	94.21
27	395	6	3	85.21	296	5	3	84.10
28	354	3	3	96.54	241	6	3	95.91
29	226	5	3	91.00	289	5	3	90.23
30	330	7	4	94.72	291	4	4	93.59
31	257	3	3	95.15	355	4	4	94.13
Average	533.29	5.16	4.03	92.41	540.77	4.58	4.00	92.09

UNIVERSITAT ROVIRA I VIRGILI
SUPERVISED AND UNSUPERVISED SEGMENTATION OF TEXTURED IMAGES BY EFFICIENT MULTI-LEVEL PATTERN CLASSIFICATION
Jaime Christian Meléndez Rodríguez
ISBN:978-84-693-7671-3/DL:T-1750-2010

Table 3.3: Classification rates (CR) for GMM-SVM-OAO configured both with "optimal" parameters and by the parameter selection algorithm described in Section 3.3.1.3. The considered parameters are: the number of SVs (SV) and the number of evaluation windows (W^*).

Image	Classifier with "optimal" parameters			Classifier with selected parameters		
	SV	W^*	CR	SV	W^*	CR
1	13132	3	98.02	13132	3	98.02
2	19435	5	96.90	19313	4	96.52
3	7775	4	98.36	7775	4	98.36
4	28228	4	95.01	28435	5	95.01
5	22075	5	83.86	22075	5	83.86
6	17056	5	96.16	17056	5	96.16
7	17195	3	91.70	17798	4	91.63
8	19220	4	92.50	19720	5	92.25
9	23234	4	94.51	23491	5	94.51
10	28565	5	95.80	28565	5	95.80
11	24255	4	89.19	24584	5	89.17
12	18505	3	84.85	20047	5	83.15
13	15726	3	96.00	16618	4	96.00
14	25934	5	73.19	25934	5	73.19
15	2237	3	93.60	1896	2	93.59
16	2237	3	92.19	1896	2	92.18
17	2237	3	92.50	1896	2	92.29
18	2451	4	89.34	1896	2	88.99
19	5096	5	93.89	4941	4	93.21
20	4941	4	90.42	4941	4	90.42
21	4667	3	96.02	4941	4	96.02
22	5376	5	91.93	4906	3	91.71
23	4906	3	89.25	4906	3	89.25
24	5644	3	95.99	5824	4	95.62
25	5594	3	95.04	5771	4	94.42
26	1475	3	95.09	1475	3	95.09
27	3335	5	82.66	3117	3	81.28
28	5984	3	94.06	5984	3	94.06
29	2755	3	90.20	2755	3	90.20
30	3967	5	90.78	3913	4	90.02
31	7033	3	93.03	7132	4	92.46
Average	11299.10	3.81	92.00	11378.48	3.81	91.76

robust classification. These results may evidence that the texture patterns in some regions of the feature space are heavily mixed. Thus, they do not allow for a voting scheme. Nevertheless, by selecting suitable sets of prototypes, classification with a single neighbor is still good as demonstrated by the high classification rates achieved, which, for instance, are comparable or even superior to the ones corresponding to the SVM-based approach (comparisons with alternative techniques are given in Chapter 5). Furthermore, in almost all cases for which several nearest neighbors have been selected, they correspond to the smallest windows, which, according to the multi-level scheme, are the least critical, as they are expected to perform the lowest number of pixel classifications. In consequence, the KNN classifier proposed in this work may become the simplest *nearest neighbor* classifier ($K_j^* = 1$) without noticeable degradation in the classification rate.

3.5.4 Second Set of Experiments

In the second set of experiments, the effects of incorporating the multi-level classification scheme of Section 3.2 have been assessed by comparing the classification rates and processing times (which are the sum of feature extraction and classification times) obtained by both single-window and multi-window versions of the three proposed classifiers. The five window sizes configured in Section 3.5.2 have been considered for the single-window versions. In addition, this set of experiments allows one to directly compare the performance of the developed classifiers according to their original formulation (i.e., without any efficiency improvement strategy).

The average results shown in Table 3.4 indicate that the proposed multi-level classification scheme is clearly superior to any of the evaluated single-window classifiers in terms of classification rates. From another perspective, it could be stated that this scheme can considerably improve the performance of a "weak" texture classifier, such as the baseline, single-window KNN and SVM-based classifiers utilized in this work. On the other hand, it certainly introduces overhead due to the refinement process

UNIVERSITAT ROVIRA I VIRGILI
SUPERVISED AND UNSUPERVISED SEGMENTATION OF TEXTURED IMAGES BY EFFICIENT MULTI-LEVEL PATTERN CLASSIFICATION
Jaime Christian Meléndez Rodríguez
ISBN:978-84-693-7671-3/DL:T-1750-2010

Table 3.4: Average classification rates ($\overline{CR}$) and average processing times ($\overline{PT}$) for the approaches evaluated in the second set of experiments.

		Evaluation window					
Classifier		3×3	5×5	9×9	17×17	33×33	Combined
GMM-RDC-KNN	$\overline{CR}$	79.71	85.32	87.38	85.97	79.57	91.75
	$\overline{PT}$	0.330	0.629	0.829	0.947	0.852	2.189
GMM-NCC-KNN	$\overline{CR}$	79.72	83.38	85.72	85.29	78.80	92.09
	$\overline{PT}$	0.395	0.592	0.625	0.741	0.797	2.211
GMM-SVM-OAO	$\overline{CR}$	83.74	87.15	87.75	86.48	81.04	91.76
	$\overline{PT}$	49.032	25.740	11.336	4.731	2.540	24.235

and, mainly, due to feature extraction, which accounts for several convolutions and averaging operations for each evaluation window size.

Nevertheless, the multi-level approach may be preferred, as it is more flexible. In fact, under realistic conditions, it would not be possible to determine which window size would yield the best classification in advance for a given input image and tune it accordingly, as there would be no ground-truth feedback. On the contrary, that would not be a problem for the proposed approach, since it naturally uses multi-sized evaluation windows for classification. Figure 3.6 illustrates some instances where the advantages of the multi-level classifiers are evident by comparing their classification maps with those produced by their associated best single window approaches, which, in all cases, correspond to the 9×9 window.

Among the multi-level classifiers, both GMM-RDC-KNN and GMM-NCC-KNN are comparable, as there is a slight difference in the classification rate in favor of GMM-NCC-KNN, but there is a slight difference in the processing time in favor of GMM-RDC-KNN. This difference in processing time may seem contradictory if these results are complemented with the information in Tables 3.1 and 3.2, as the number of prototypes for GMM-NCC-KNN is lower than the one for GMM-RDC-

UNIVERSITAT ROVIRA I VIRGILI
SUPERVISED AND UNSUPERVISED SEGMENTATION OF TEXTURED IMAGES BY EFFICIENT MULTI-LEVEL PATTERN CLASSIFICATION
Jaime Christian Meléndez Rodríguez
ISBN:978-84-693-7671-3/DL:T-1750-2010

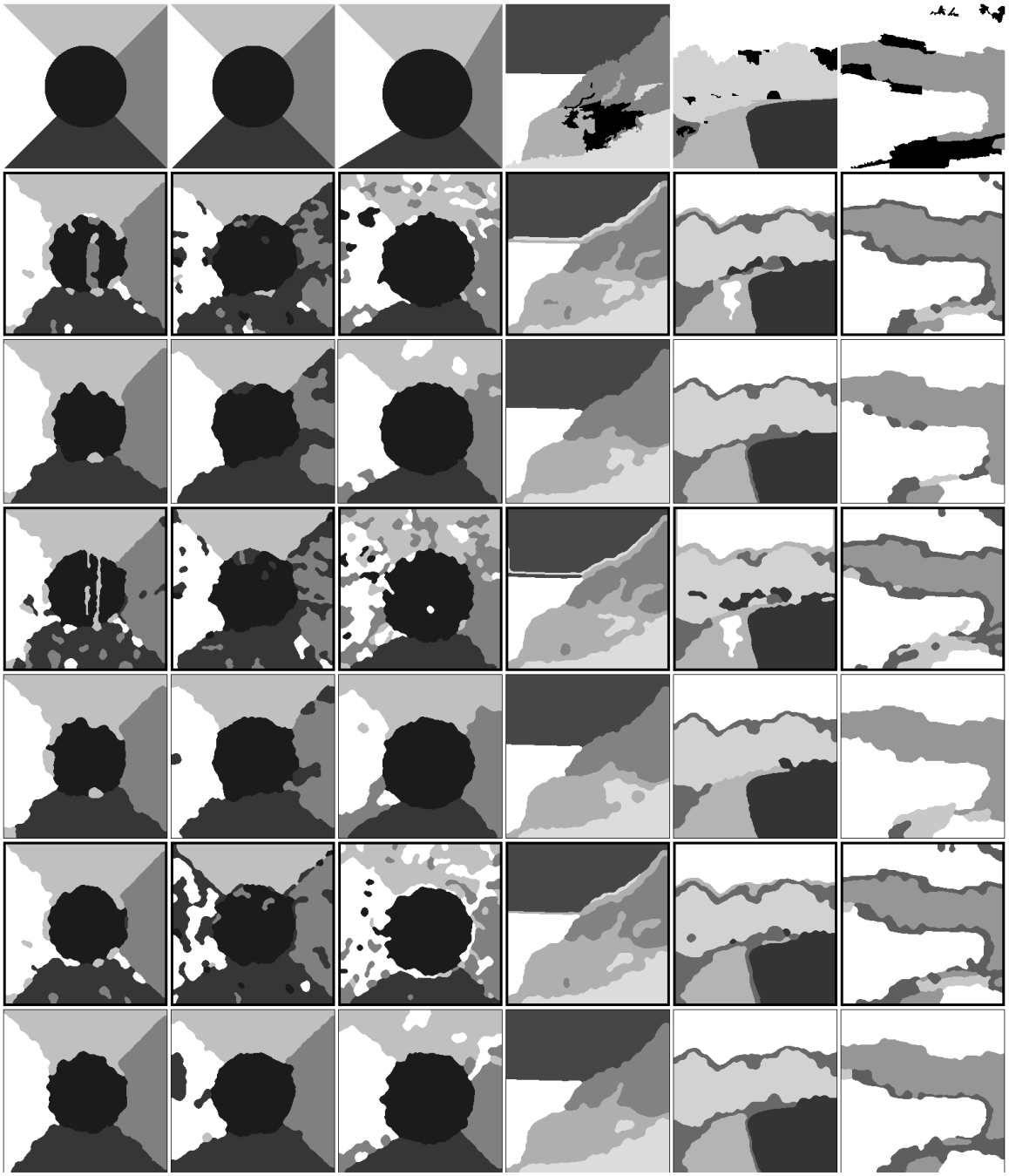

Figure 3.6: Classification maps for images 6, 8, 11, 19, 22 and 25. First row: ground-truth. Starting from the second row, results by: GMM-RDC-KNN (9×9 window), GMM-RDC-KNN (multi-level), GMM-NCC-KNN (9×9 window), GMM-NCC-KNN (multi-level), GMM-SVM-OAO (9×9 window) and GMM-SVM-OAO (multi-level).

KNN. However, the number of evaluation windows for the former is greater and, as stated above, it has more influence on the processing time due to feature extraction. Regarding GMM-SVM-OAO, although its classification rate is similar to the other two, its processing time is significantly higher due to the large number of SVs.

3.5.5 Third Set of Experiments

In the third set of experiments, the prototypes determined by the RDC and NCC algorithms utilized in this work have been compared with those determined by representatives of the alternative clustering algorithms discussed in Sections 2.3.1.1 (*K-Nearest Neighbors*) and 3.3.1 in terms of classification accuracy, number of elements and classification time. The following alternative algorithms have been taken into account: k-means, g-means, mean shift and learning vector quantization (LVQ). For completeness, the extreme classifier that stores all the available feature vectors corresponding to the training patterns as prototypes has also been included. While g-means can automatically determine an "optimal" number of clusters, there is no optimality criterion to set the number of clusters for k-means, the bandwidth for mean shift or the number of neurons for the LVQ networks. Therefore, in order to set these and the remaining parameters of the classifier, the parameter selection algorithm of Section 3.3.1.3 has been applied.

The performance of the evaluated approaches is summarized in Table 3.5. Both RDC and NCC yield higher classification rates than the remaining clustering approaches. Although the difference in the classification rate is small compared to k-means and g-means, the difference in the number of prototypes is significant, especially in the latter case, which is probably related to the dependence of g-means on the shape of the clusters, which makes the algorithm to partition the feature space into finer levels in order to meet the Gaussian hypothesis. The contrary occurs if LVQ is now considered, as it yields a similar number of prototypes, but they lack the accuracy of those determined by RDC and NCC, even though LVQ takes into account the

UNIVERSITAT ROVIRA I VIRGILI
SUPERVISED AND UNSUPERVISED SEGMENTATION OF TEXTURED IMAGES BY EFFICIENT MULTI-LEVEL PATTERN CLASSIFICATION
Jaime Christian Meléndez Rodríguez
ISBN:978-84-693-7671-3/DL:T-1750-2010

Table 3.5: Average classification rate ($\overline{CR}$), average number of prototypes ($\overline{P^*}$) and average classification time ($\overline{CT}$) for the approaches evaluated in the third set of experiments.

Clustering approach	$\overline{CR}$	$\overline{P^*}$	$\overline{CT}$
all feature vectors	91.83	824793.20	730.880
RDC	91.75	577.65	0.794
NCC	92.09	540.77	0.704
k-means	91.39	760.70	0.974
g-means	91.23	1206.30	1.504
mean shift	88.81	772.77	1.382
LVQ	86.72	548.83	0.725

interaction between feature vectors of different classes instead of processing samples of each class separately, as the remaining approaches do.

Compared to mean shift, the results are clearly favorable according to all the performance criteria. It is worth to note that mean shift achieves a lower classification rate than k-means, which may confirm the previous claim by Jurie and Triggs (2005) mentioned in Section 3.3.1.1. Furthermore, even when both algorithms have almost the same number of prototypes, classification with mean shift takes much more time. The first reason is that mean shift sometimes produces moderately large numbers of prototypes even for the largest windows, which, on the contrary, should be the ones with the fewest prototypes, as they have more information about the texture being analyzed and, thus, have more discrimination capabilities. Since these windows are expected to perform most of the pixel classifications according to the top-down scheme, the scheme's efficiency is not well-spent. The second reason is that classification with mean shift prototypes introduces some false boundaries that must be refined in the subsequent levels. Therefore, more pixel classifications are carried out.

The last point to remark is that, contrarily to what would be expected, using all the available feature vectors as prototypes does not always produce the best classi-

fication accuracy. Probably, the main reason for the lower average score obtained by this approach is the "curse of dimensionality", although applied to the excess of prototypes instead of to the excess of features. In this sense, since there is a prototype for each training pixel, due to the noisy nature of texture, there are cases where the resulting classifier becomes too specific and is not able to discriminate between patterns that, locally, are quite similar, but, globally, are indeed different. On the other hand, by only considering a set of representative prototypes, better generalization and abstraction is achieved. Obviously, in the remaining cases, using all feature vectors as prototypes yields better classification accuracy. However, by taking into account the associated computation time, it may not be a good choice depending on the application. Therefore, in such situations, both RDC and NCC are the best tradeoffs, as demonstrated by the results shown in Table 3.5.

In order to provide complementary information and visual feedback for this set of experiments, Figures 3.7 and 3.8 show some classification maps comparing the top performing approaches according to classification rate. Figure 3.7 corresponds to cases where the extreme classifier fed with all feature vectors as prototypes is surpassed by both of the proposed techniques. They are mostly instances belonging to the outdoor scenes. Figure 3.8 shows examples of the remaining cases.

3.5.6 Fourth Set of Experiments

The fourth set of experiments compares the performance of different kernels for the SVM-based classifier, as it is usually argued that non-linear kernels have more discrimination capabilities than the linear kernel chosen in this work (e.g., Kim et al., 2002; Park and Park, 2007), which, on the other hand, is necessary in order to apply the SV reduction method of Section 3.4.2. Alternative evaluated kernels correspond to: the polynomial kernel with degrees 2 to 5, the radial basis function (RBF) kernel and the sigmoid (or neural) kernel. Comparisons are carried out in terms of classification rate, number of SVs and classification time.

UNIVERSITAT ROVIRA I VIRGILI
SUPERVISED AND UNSUPERVISED SEGMENTATION OF TEXTURED IMAGES BY EFFICIENT MULTI-LEVEL PATTERN CLASSIFICATION
Jaime Christian Meléndez Rodríguez
ISBN:978-84-693-7671-3/DL:T-1750-2010

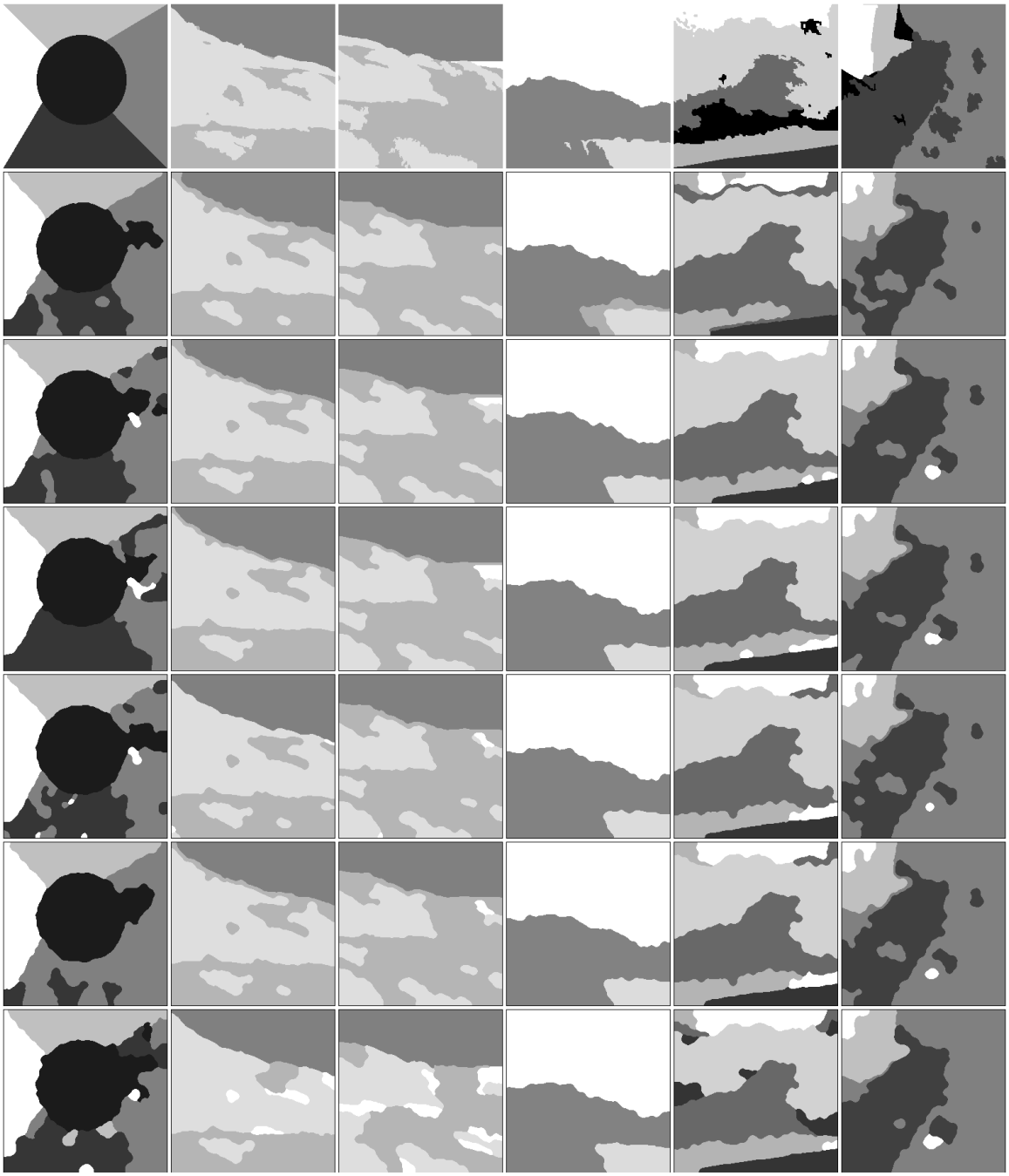

Figure 3.7: Classification maps for images 5, 17, 18, 21, 23 and 27. First row: ground-truth. Starting from the second row, results by the KNN classifier with: all feature vectors as prototypes, RDC, NCC, k-means, g-means and mean shift.

UNIVERSITAT ROVIRA I VIRGILI
SUPERVISED AND UNSUPERVISED SEGMENTATION OF TEXTURED IMAGES BY EFFICIENT MULTI-LEVEL PATTERN CLASSIFICATION
Jaime Christian Meléndez Rodríguez
ISBN:978-84-693-7671-3/DL:T-1750-2010

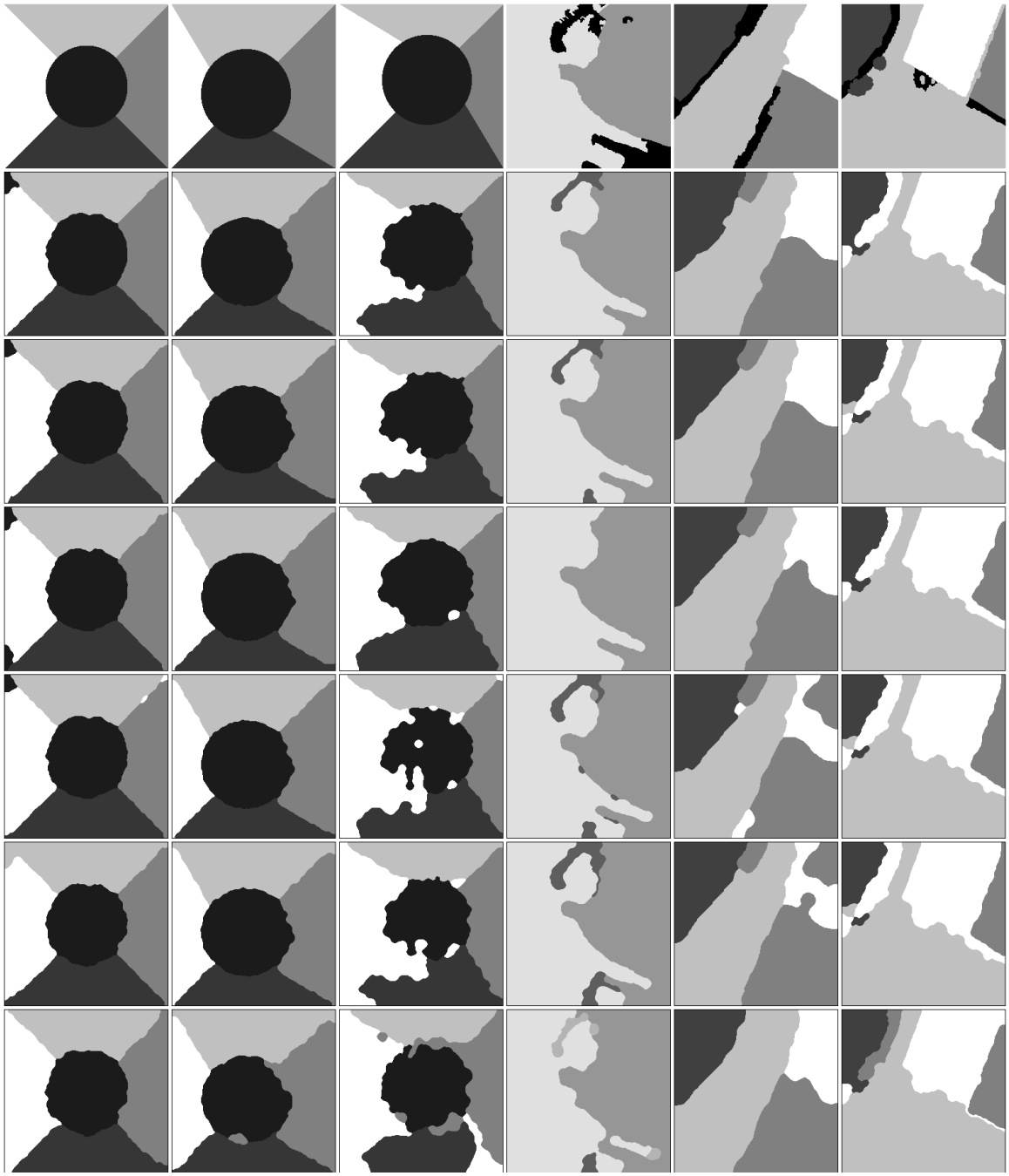

Figure 3.8: Classification maps for images 1, 3, 4, 24, 26 and 30. First row: ground-truth. Starting from the second row, results by the KNN classifier with: all feature vectors as prototypes, RDC, NCC, k-means, g-means and mean shift.

UNIVERSITAT ROVIRA I VIRGILI
SUPERVISED AND UNSUPERVISED SEGMENTATION OF TEXTURED IMAGES BY EFFICIENT MULTI-LEVEL PATTERN CLASSIFICATION
Jaime Christian Meléndez Rodríguez
ISBN:978-84-693-7671-3/DL:T-1750-2010

Table 3.6: Average classification rate ($\overline{CR}$), average number of SVs ($\overline{SV}$) and average classification time ($\overline{CT}$) for the approaches evaluated in the fourth set of experiments.

Kernel	$\overline{CR}$	$\overline{SV}$	$\overline{CT}$
linear	91.76	11378.48	22.840
polynomial (deg. 2)	88.25	37886.17	130.480
polynomial (deg. 3)	68.66	52726.73	212.068
polynomial (deg. 4)	44.81	60120.03	208.981
polynomial (deg. 5)	27.87	62193.53	159.759
RBF	91.96	21613.53	64.958
sigmoid	91.53	26336.00	123.949

The results of this comparison are shown in Table 3.6. As expected, the RBF kernel yields the best classification rate. However, it is only 0.2 points higher than the one achieved by the linear kernel. More important, the SVM optimization algorithm determines two times more SVs for the RBF kernel than for the linear kernel, and its classification time is three times greater. Thus, the linear kernel is not only mandatory for SV reduction, but the best choice in terms of efficiency.

Another point to remark is the performance of the polynomial kernel, which significantly degrades as the degree of the polynomial increases. Probably, this is due to the one-against-one scheme, which may be unsuitable for very high-dimensional spaces (higher the degree, higher the dimensionality), since contradictory results are reported in (Kim et al., 2002), where the best classification corresponds to the polynomial kernel with degree five using the one-against-all extension. On the other hand, the mapping of the polynomial kernel may be decreasing the separability among classes as the degree increases, which becomes apparent by the high and increasing number of resulting SVs, which, by definition, are the feature vectors closest to the opposite class.

UNIVERSITAT ROVIRA I VIRGILI
SUPERVISED AND UNSUPERVISED SEGMENTATION OF TEXTURED IMAGES BY EFFICIENT MULTI-LEVEL PATTERN CLASSIFICATION
Jaime Christian Meléndez Rodríguez
ISBN:978-84-693-7671-3/DL:T-1750-2010

3.5.7 Fifth Set of Experiments

In the last set of experiments, the efficiency improvements proposed in Section 3.4 are validated by comparing the new resulting performance with the one achieved by the original classifiers described in Section 3.3 and that has already been evaluated along the previous experiments. In addition, results obtained by applying those strategies with the original heuristic developed in (Puig and Garcia, 2006) are also included. These comparisons have been carried out in three parts.

First Part

In the first part, the strategies aiming at reducing the number of prototypes (Section 3.4.1) or SVs (Section 3.4.2) are considered. The performance criteria for this part are: the classification rate, the number of prototypes or SVs, and the classification time. The classifiers that follow the proposed strategies are referred to as GMM-RDC-KNN+PR, GMM-NCC-KNN+PR and GMM-SVM-OAO+SVR, while the classifiers based on the original heuristic described in (Puig and Garcia, 2006) are referred to as GMM-RDC-KNN+PR′ and GMM-NCC-KNN+PR′.

The results corresponding to the first part are summarized in Table 3.7. They show that with the prototype reduction strategy, more than half of the previously determined prototypes are discarded, which yields a similar decrease in the computation time, while preserving the classification rates or even slightly improving them as in the case of GMM-RDC-KNN+PR. In fact, this classifier is the best among the prototype-based approaches in general terms.

The results in Table 3.7 also indicate that the modified version of the performance heuristic originally developed by Puig and Garcia (2006) always yields fewer prototypes than the latter, which is more evident for the RDC-based algorithm. In addition, it yields slightly higher classification rates, even when the original heuristic aims at maximizing the benefit more than its modified version, as discussed in Section 3.4.1. The main reason is that, while the cost versus benefit analysis is carried

UNIVERSITAT ROVIRA I VIRGILI
SUPERVISED AND UNSUPERVISED SEGMENTATION OF TEXTURED IMAGES BY EFFICIENT MULTI-LEVEL PATTERN CLASSIFICATION
Jaime Christian Meléndez Rodríguez
ISBN:978-84-693-7671-3/DL:T-1750-2010

Table 3.7: Average classification rate ($\overline{CR}$), average number of prototypes ($\overline{P^*}$) or SVs ($\overline{SV}$), and average classification time ($\overline{CT}$) for the approaches evaluated in the first part of the fifth set of experiments.

Classifier	$\overline{CR}$	$\overline{P^*}/\overline{SV}$	$\overline{CT}$
GMM-RDC-KNN	91.75	577.65	0.794
GMM-RDC-KNN+PR	91.86	208.47	0.324
GMM-RDC-KNN+PR′	91.72	258.13	0.402
GMM-NCC-KNN	92.09	540.77	0.704
GMM-NCC-KNN+PR	91.71	228.73	0.392
GMM-NCC-KNN+PR′	91.70	248.30	0.419
GMM-SVM-OAO	91.76	11378.48	22.840
GMM-SVM-OAO+SVR	91.76	32.87	0.296

out on single windows, the evaluation of the resulting prototypes is carried out over the proposed top-down multi-window approach. Therefore, while descending through the classification levels, the possible loss in accuracy due to the reduction of the number of prototypes is usually minimized, since fewer and fewer pixels are reclassified at the subsequent levels. Figure 3.9 shows classification maps for the prototype-based approaches by considering representatives of the different image sets.

Regarding the SV reduction method, the improvement in classification time is of two orders of magnitude, which makes GMM-SVM-OAO+SVR the fastest classifier. The pixel classification results are exactly the same with or without SV reduction.

Second Part

In the second part, only feature selection is considered (Section 3.4.3). The performance criteria are: the classification rate, the number of processed features, the feature extraction time and the classification time. While feature selection does not affect the prototype sets computed for the KNN classifiers, it does affect the SVs for the SVM-based classifier, since the latter is more sensitive and must be retrained

UNIVERSITAT ROVIRA I VIRGILI
SUPERVISED AND UNSUPERVISED SEGMENTATION OF TEXTURED IMAGES BY EFFICIENT MULTI-LEVEL PATTERN CLASSIFICATION
Jaime Christian Meléndez Rodríguez
ISBN:978-84-693-7671-3/DL:T-1750-2010

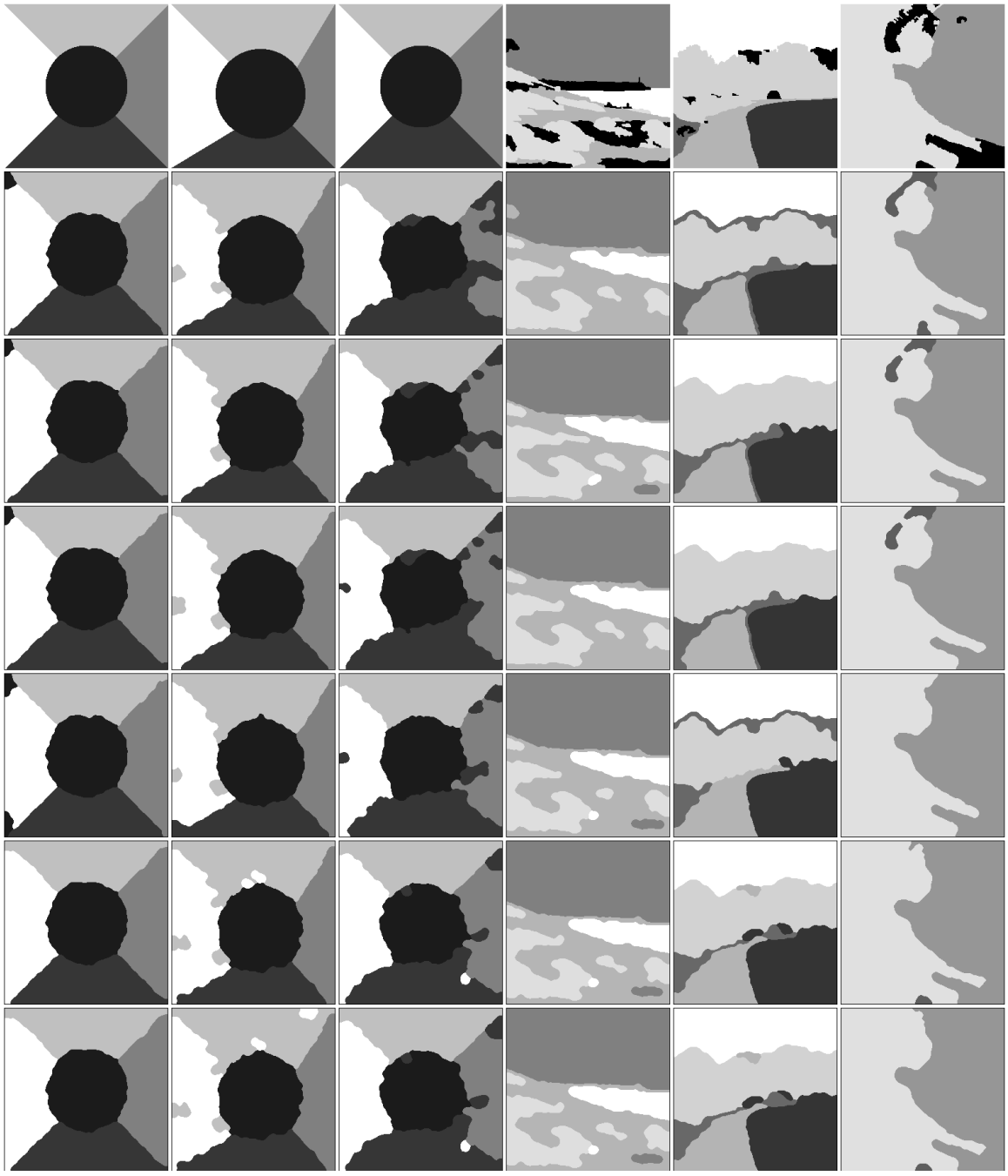

Figure 3.9: Classification maps for images 1, 2, 8, 15, 22 and 24. First row: ground-truth. Starting from the second row, results by: GMM-RDC-KNN, GMM-RDC-KNN+PR, GMM-RDC-KNN+PR′, GMM-NCC-KNN, GMM-NCC-KNN+PR and GMM-NCC-KNN+PR′.

UNIVERSITAT ROVIRA I VIRGILI
SUPERVISED AND UNSUPERVISED SEGMENTATION OF TEXTURED IMAGES BY EFFICIENT MULTI-LEVEL PATTERN CLASSIFICATION
Jaime Christian Meléndez Rodríguez
ISBN:978-84-693-7671-3/DL:T-1750-2010

Table 3.8: Average classification rate ($\overline{CR}$), average number of prototypes ($\overline{P^*}$) or SVs ($\overline{SV}$), average number of processed features ($\overline{M^*}$), average feature extraction time ($\overline{FET}$), average classification time ($\overline{CT}$) and average processing time ($\overline{PT}$) for the approaches evaluated in the second part of the fifth set of experiments.

Classifier	$\overline{CR}$	$\overline{P^*}/\overline{SV}$	$\overline{M^*}$	$\overline{FET}+\overline{CT}=\overline{PT}$
GMM-RDC-KNN	91.75	577.65	180.80	1.395+0.794=2.189
GMM-RDC-KNN+FS	91.74	577.65	50.33	0.490+0.257=0.747
GMM-RDC-KNN+FS′	91.79	577.65	57.20	0.565+0.310=0.875
GMM-NCC-KNN	92.09	540.77	190.40	1.507+0.704=2.211
GMM-NCC-KNN+FS	91.11	540.77	57.10	0.546+0.260=0.806
GMM-NCC-KNN+FS′	91.40	540.77	66.83	0.644+0.289=0.933
GMM-SVM-OAO	91.76	11378.48	180.80	1.395+22.840=24.235
GMM-SVM-OAO+FS	90.97	13628.10	50.33	0.499+12.549=13.048
GMM-SVM-OAO+FS′	91.06	13421.10	61.50	0.609+14.001=14.610

for each new feature subset in order to perform well. In consequence, the number of SVs are also reported and does not necessarily correspond to the one in the previous part. The classifiers based on the proposed feature selection method are referred to as GMM-RDC-KNN+FS, GMM-NCC-KNN+FS and GMM-SVM-OAO+FS. In turn, the classifiers based on the original heuristic are referred to as GMM-RDC-KNN+FS′, GMM-NCC-KNN+FS′ and GMM-SVM-OAO+FS′.

Table 3.8 summarizes the obtained results. Again, a considerable impact on the processing time is observed, which is even more significant than in the previous experiments, since feature selection affects both feature extraction and classification. However, more degradation in the classification rate is undergone for the last two classifiers. Nevertheless, this degradation is around 1% in the worst case (GMM-NCC-KNN+FS), but it is well compensated by the improvement of 63% in the processing time. Figure 3.10 compares the classification maps obtained by the proposed classifiers with and without feature selection for the same images as in Figure 3.9.

UNIVERSITAT ROVIRA I VIRGILI
SUPERVISED AND UNSUPERVISED SEGMENTATION OF TEXTURED IMAGES BY EFFICIENT MULTI-LEVEL PATTERN CLASSIFICATION
Jaime Christian Meléndez Rodríguez
ISBN:978-84-693-7671-3/DL:T-1750-2010

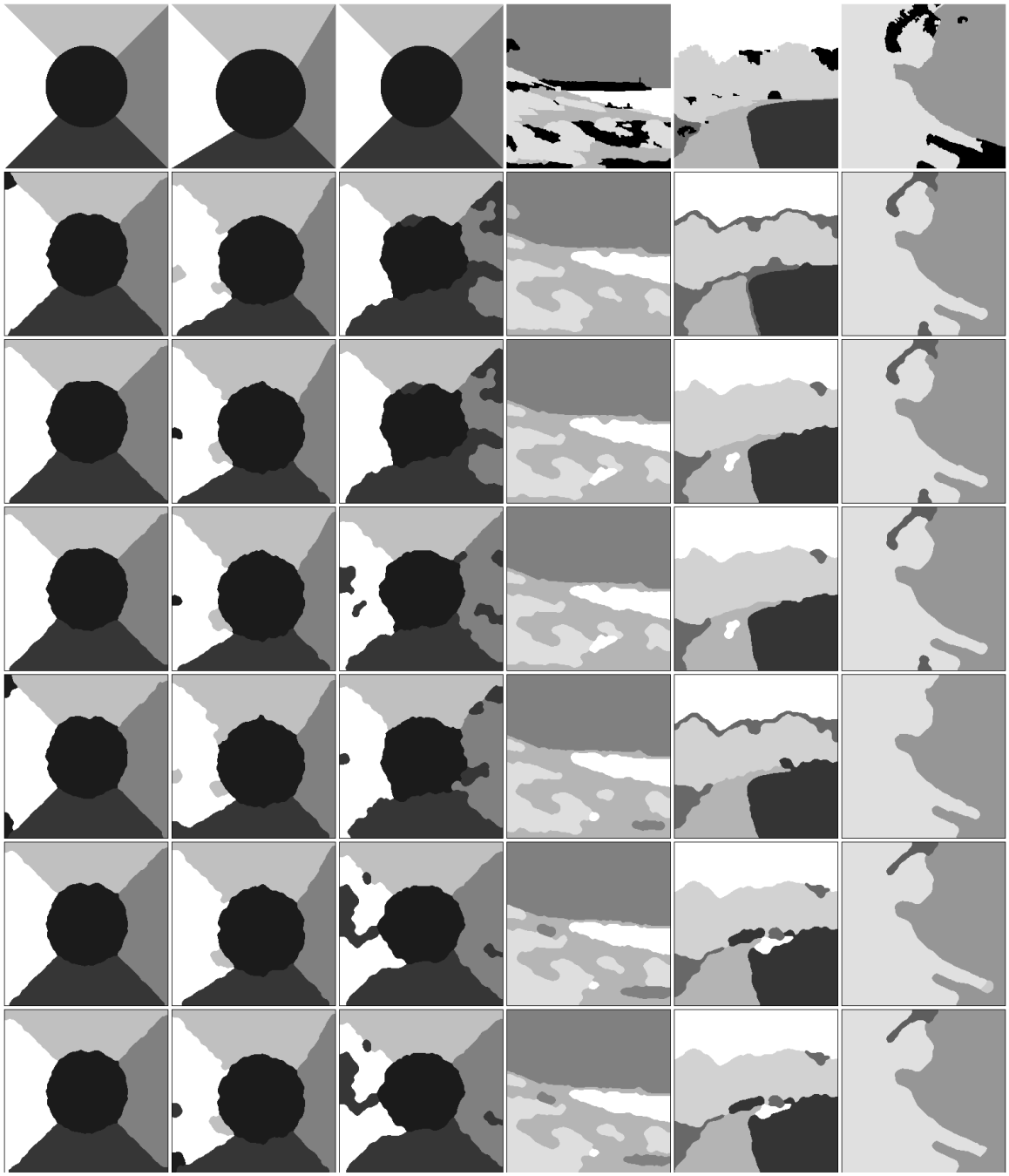

Figure 3.10: Classification maps for images 1, 2, 8, 15, 22 and 24. First row: ground-truth. Starting from the second row, results by: GMM-RDC-KNN, GMM-RDC-KNN+FS, GMM-RDC-KNN+FS′, GMM-NCC-KNN, GMM-NCC-KNN+FS and GMM-NCC-KNN+FS′.

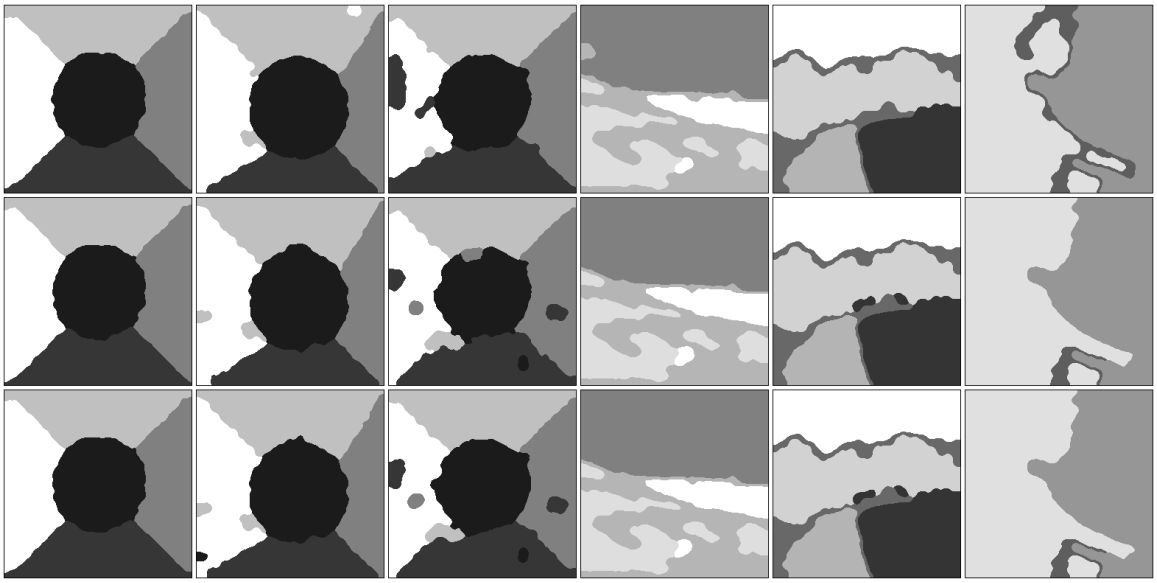

Figure 3.10: Classification maps for images 1, 2, 8, 15, 22 and 24 (continued). In order from the first row, results by: GMM-SVM-OAO, GMM-SVM-OAO+FS, GMM-SVM-OAO+FS′.

As in the previous part, the modified performance heuristic is more effective than the original one for the same reason given before for the prototype reduction strategy. Although the difference is more apparent for the RDC-based classifiers, in the remaining cases, it is very advantageous as well, as the reduction in the number of features and processing time is more significant than the decrease in the classification accuracy. For instance, around a 14% reduction in processing time versus a 0.3% degradation in classification accuracy is observed for the NCC-based classifiers. For the SVM-based classifiers, these numbers change to 10% versus 0.1%, respectively.

Third Part

In the last part, both reduction of prototypes/SVs and feature selection are evaluated together. The performance criteria are the combination of those utilized in the previous two parts. The classifiers based on the proposed strategies are referred to as GMM-RDC-KNN+PR+FS, GMM-NCC-KNN+PR+FS and GMM-

UNIVERSITAT ROVIRA I VIRGILI
SUPERVISED AND UNSUPERVISED SEGMENTATION OF TEXTURED IMAGES BY EFFICIENT MULTI-LEVEL PATTERN CLASSIFICATION
Jaime Christian Meléndez Rodríguez
ISBN:978-84-693-7671-3/DL:T-1750-2010

Table 3.9: Average classification rate ($\overline{CR}$), average number of prototypes ($\overline{P^*}$) or SVs ($\overline{SV}$), average number of processed features ($\overline{M^*}$), average feature extraction time ($\overline{FET}$), average classification time ($\overline{CT}$) and average processing time ($\overline{PT}$) for the approaches evaluated in the third part of the fifth set of experiments.

Classifier	$\overline{CR}$	$\overline{P^*}/\overline{SV}$	$\overline{M^*}$	$\overline{FET}+\overline{CT}=\overline{PT}$
GMM-RDC-KNN	91.75	577.65	180.80	1.395+0.794=2.189
GMM-RDC-KNN+PR+FS	91.79	208.47	55.37	0.538+0.179=0.717
GMM-RDC-KNN+PR′+FS′	91.30	258.13	58.90	0.554+0.204=0.758
GMM-NCC-KNN	92.09	540.77	190.40	1.507+0.704=2.211
GMM-NCC-KNN+PR+FS	91.24	228.73	60.03	0.573+0.186=0.759
GMM-NCC-KNN+PR′+FS′	91.31	248.30	69.87	0.668+0.213=0.881
GMM-SVM-OAO	91.76	11378.48	180.80	1.395+22.840=24.235
GMM-SVM-OAO+SVR+FS	90.97	32.87	50.33	0.499+0.139=0.638
GMM-SVM-OAO+SVR+FS′	91.06	32.87	61.50	0.609+0.149=0.758

SVM-OAO+SVR+FS. On the other hand, the classifiers based on the original heuristic in both strategies are referred to as GMM-RDC-KNN+PR′+FS′, GMM-NCC-KNN+PR′+FS′ and GMM-SVM-OAO+SVR+FS′.

The results for this part are shown in Table 3.9. By combining all the strategies evaluated above, the most efficient versions of the proposed classifiers are obtained. Moreover, the trends observed in the previous experiments are also observed here. For instance, the RDC-based classifier with the modified performance heuristic for prototype reduction and feature selection (GMM-RDC-KNN+PR+FS) is the one with the highest classification rate, while its SVM-based equivalent (GMM-SVM-OAO+SVR+FS) is the fastest.

In some cases, degradation in the classification rate is experienced. However, it is small and not as significant as the reduction in the processing time. As a visual feedback, Figure 3.11 compares the classification maps yielded by the original proposed classifiers and their efficiency-improved versions for the same images as in

Figure 3.9 (the classification maps for the SVM-based approaches are not shown since they are the same as in Figure 3.10).

Finally, the modified performance heuristic applied to the combination of prototype reduction and feature selection yields again more efficient approaches than its original version in (Puig and Garcia, 2006), as it always selects a lower number of elements in both categories with almost the same or even a better classification accuracy.

3.6 Summary

This chapter describes a new technique for supervised texture segmentation based on a supervised pixel-based texture classifier that utilizes an efficient multi-level pattern classification algorithm in order to process texture features obtained after evaluating a set of texture methods over multiple windows of different size. The multi-level classification algorithm follows a top-down approach that first classifies every image pixel with the largest available evaluation window and then proceeds with the smaller ones, each time only reclassifying those pixels located in boundary zones. The pixel-based classification in the proposed technique is carried out by two alternative pattern recognition methods: KNN and SVMs.

For the KNN classifier, the main issue is determining appropriate sets of prototypes in order to model the texture classes of interest. Two approaches relying on clustering algorithms have been proposed. The first approach consists of a resolution-driven clustering algorithm that continuously bipartitions the existing clusters according to a resolution parameter related to their size. The second method exploits the optimality of the normalized cut criterion in order to partition the feature space. In both cases, prototypes are associated with the centroids of the resulting clusters.

For the SVM-based classifier, a one-against-one multiclass extension of binary SVMs is utilized. The final classification result is obtained after applying a majority

UNIVERSITAT ROVIRA I VIRGILI
SUPERVISED AND UNSUPERVISED SEGMENTATION OF TEXTURED IMAGES BY EFFICIENT MULTI-LEVEL PATTERN CLASSIFICATION
Jaime Christian Meléndez Rodríguez
ISBN:978-84-693-7671-3/DL:T-1750-2010

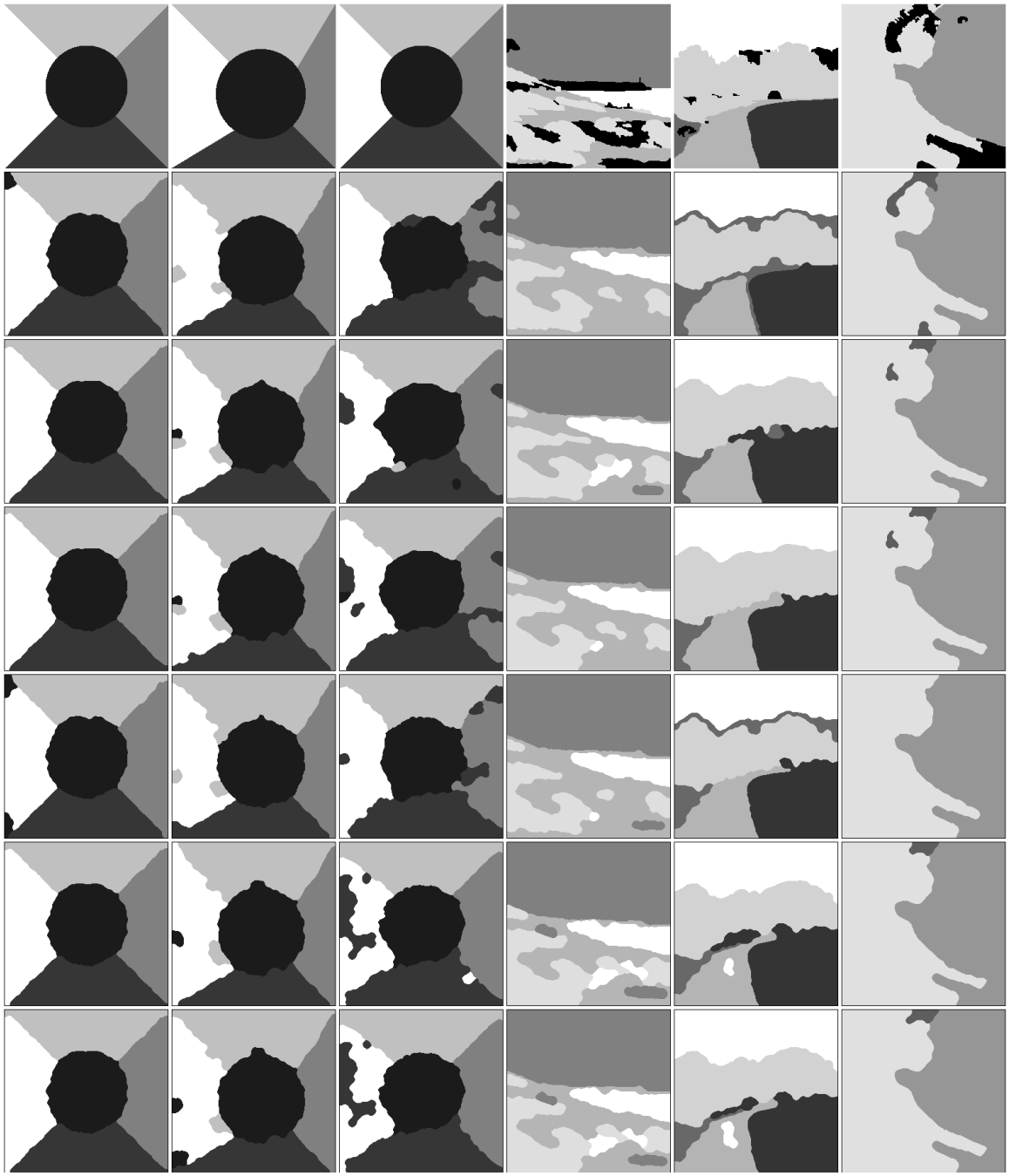

Figure 3.11: Classification maps for images 1, 2, 8, 15, 22 and 24. First row: ground-truth. Starting from the second row, results by: GMM-RDC-KNN, GMM-RDC-KNN+PR+FS, GMM-RDC-KNN+PR′+FS′, GMM-NCC-KNN, GMM-NCC-KNN+PR+FS and GMM-NCC-KNN+PR′+FS′.

voting rule. A modified version of the original discrete voting scheme that uses probability estimates instead of the sign of the decision function has been included.

In order to configure the developed classifiers with suitable parameters, a parameter selection algorithm has also been proposed. It performs a search for the best combination of parameters considering the classification rates obtained after classifying the pixels of the training texture samples. It also determines the number of evaluation windows necessary to discriminate among textures according to a threshold for the minimum, acceptable classification performance. While the complete algorithm is applied to the KNN classifier, only the part related to the necessary number of evaluation windows is applied to the SVM-based classifier.

The proposed classifiers are enhanced by including three additional strategies for improving their efficiency. The first strategy reduces the number of prototypes evaluated by the KNN classifier by applying a heuristic performance measure that aims at maximizing the classification rate while minimizing the cost. The second strategy summarizes all the available information of each of the SVMs that make up the multiclass SVM-based classifier into a single SV by aggregating the individual SVs and the coefficients that result after regular optimization. In this case, it is mandatory to use linear kernels for configuring the classifier. The third strategy performs feature selection by extending the heuristic performance measure applied for prototype reduction. A sequential forward generation procedure is used in order to accomplish this extension.

Several sets of experiments with the developed classifiers have been carried out. First, the effectiveness of the algorithm for configuring the parameters of these techniques has been assessed by comparing the performance of the classifiers configured by this algorithm with the best performance resulting after classifying the test set with different configurations. The results obtained with parameters determined by the proposed parameter selection algorithm are very close to the best ones. Then, improvements in classification quality due to the proposed multi-level classification

scheme have been demonstrated by comparing the classification rates of the multi-window classifiers with the classification rates of their single-window counterparts.

In the next two sets of experiments, alternative clustering algorithms for determining sets of prototypes for the KNN classifier, as well as alternative kernel functions for the SVM-based classifier, have been evaluated. In the first case, the results indicate that both of the proposed methods are the best choice in considering their higher accuracy and their lower number of prototypes. Furthermore, the classifier with prototypes determined by the normalized cut clustering algorithm yields a higher average classification rate than the extreme approach that utilizes all the training feature vectors as prototypes. In the second case, the simple linear kernel is shown to yield a slightly lower classification rate than the best performer among the more sophisticated kernels, but with a significantly lower number of SVs and classification time.

In the last set of experiments, the proposed classifiers are compared with their efficiency-improved counterparts, which achieve very close or, sometimes, superior classification accuracy with a considerable reduction in the number of evaluated prototypes, SVs and features, which leads to equivalent savings in processing time. Moreover, the results indicate that the proposed performance heuristic yields more efficient approaches than the original heuristic proposed by Puig and Garcia (2006) in all the experiments.

UNIVERSITAT ROVIRA I VIRGILI
SUPERVISED AND UNSUPERVISED SEGMENTATION OF TEXTURED IMAGES BY EFFICIENT MULTI-LEVEL PATTERN CLASSIFICATION
Jaime Christian Meléndez Rodríguez
ISBN:978-84-693-7671-3/DL:T-1750-2010

UNIVERSITAT ROVIRA I VIRGILI
SUPERVISED AND UNSUPERVISED SEGMENTATION OF TEXTURED IMAGES BY EFFICIENT MULTI-LEVEL PATTERN CLASSIFICATION
Jaime Christian Meléndez Rodríguez
ISBN:978-84-693-7671-3/DL:T-1750-2010

Chapter 4

Unsupervised Texture Segmentation

This chapter presents a new unsupervised segmentation technique that is an extension to the unsupervised domain of the supervised pixel-based classifier described in Chapter 3. This chapter is organized as follows. Section 4.1 introduces the problem of unsupervised texture segmentation. Section 4.2 describes the proposed unsupervised segmentation technique and its main stages. Section 4.3 details the application of the proposed methodology and shows and discusses three sets of experiments. The contents of the chapter are finally summarized in Section 4.4.

4.1 Introduction

Image segmentation consists of partitioning a given image into disjoint regions of uniform features. It is a complex task, since it requires the detection of those features within each region, as well as the location of the boundaries that separate the different regions. Segmentation is usually a preliminary stage of further processing and analysis tasks, such as classification and interpretation. In order to segment an image, it is necessary to extract features and derive measures from them that enable

the segregation of the distinctive regions contained in the image. Texture is one of the most important features utilized for image segmentation.

Texture segmentation can be either supervised or unsupervised, depending on whether prior knowledge about the image or its texture classes is available or not. *Supervised texture segmentation* identifies and separates regions that match texture properties previously learnt in training samples (Chapter 3). In turn, *unsupervised texture segmentation* discriminates the texture classes of the image, as well as separates them into regions. Compared to supervised segmentation, the unsupervised case is more flexible, although more challenging.

Despite the success of many of the unsupervised texture segmentation algorithms proposed in the literature, their supervised counterparts usually perform better in terms of segmentation accuracy as demonstrated in previous works (e.g., Garcia and Puig, 2003). The reason is that, by definition, unsupervised texture segmentation algorithms do not take any advantage of any prior knowledge concerning the texture patterns to be discriminated, since this kind of information is not available. On the contrary, supervised algorithms are specifically trained in order to identify a set of patterns. Therefore, they are more likely to succeed, especially when those patterns are difficult to be separated.

This thesis proposes a two-stage unsupervised texture segmentation technique that aims at incorporating the aforementioned specificity by stacking a supervised classifier on top of an unsupervised segmentation algorithm. In this way, during the first stage, a clustering algorithm is used to discover the texture patterns of a given image. In the second stage, a supervised classifier trained with those patterns is used to perform pixel-based classification of the input image, yielding the final segmentation. The following sections give further details of this technique.

4.2 Unsupervised Segmentation Methodology

The unsupervised texture segmentation methodology proposed in this thesis is as follows. During the initial texture feature extraction step, a given image is processed by a set of texture methods, thus obtaining a feature vector for every pixel. Since we aim at extending the supervised classification scheme developed in the previous chapter, texture feature extraction is carried out as described in Section 3.2.

Then, a clustering algorithm is used to determine which texture patterns are present in the given image. This is referred to as *pattern discovery*. Since the objective of pattern discovery is only to find out a set of suitable patterns and not to accurately define the image regions, this process is carried out using a small number of feature vectors, thus significantly reducing the computational cost. These vectors are selected by uniform sampling of their associated image pixels. After this stage, the pixels associated with the feature vectors are resampled in order to build a labeled image (*pattern image*) of the same size as the original.

From this pattern image, a number of pixels corresponding to each texture pattern are again selected by uniform sampling, although avoiding pixels close to boundaries between regions. This contributes to selecting pixels assumed to belong to "pure" texture patterns. Next, the feature vectors associated with the selected pixels are used to train a supervised classifier, which produces the final segmented image after performing pixel-based classification by considering the complete set of feature vectors.

The pixel-based classification methodology follows the top-down approach with multi-sized evaluation windows described in Section 3.2. Since the classes that make up the classified image are spatially updated after each window size is applied, pixels selected as training samples for the next classification level are updated as well. In cases where it is not possible to take enough training samples for a given class, since all of its associated pixels are in a boundary zone, the class is discarded and not taken

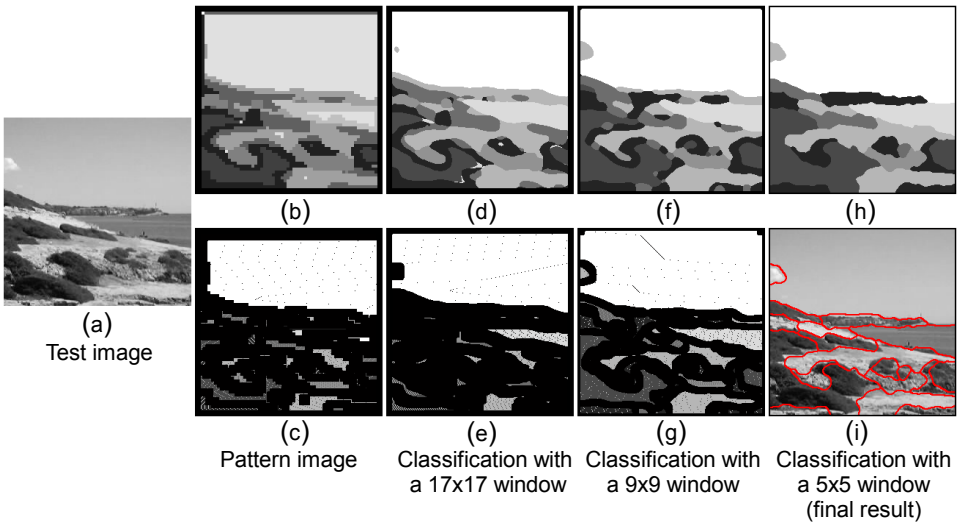

Figure 4.1: Steps performed by the proposed unsupervised segmentation technique in order to process a given image by extending the supervised pixel-based classification scheme developed in Chapter 3.

into account in the remaining classification levels.

Figure 4.1 shows how the proposed technique processes a given image, Figure 4.1(a), considering three evaluation window sizes ($W = 3$). First, feature vectors associated with the image are clustered and resampled producing an initial segmentation composed by the texture patterns that will be used as models for the classification stage (Figure 4.1(b)). Then, pixels belonging to frontier regions in that approximate segmentation are marked in order to avoid them when selecting training samples (Figure 4.1(c)). Pixels selected as training samples for the first classification level are also shown in this image; they are represented by complementary gray levels with respect to their corresponding patterns.

Next, the first classification level, which utilizes the largest available evaluation window, is performed and produces the result in Figure 4.1(d). The frontiers between textured regions and their neighborhoods (black regions in Figure 4.1(e)) are labeled as unknown and will be reclassified in the subsequent level. Since these unknown pixels coincide with the pixels that must be avoided as training samples, the same

procedure is applied to both tasks. Again, pixels selected as training samples are shown in complementary gray levels. Next, those unknown pixels are reclassified with the next window size (Figure 4.1(f)), and the process is iterated until the image is reclassified with the smallest available window, yielding the final segmentation, Figures 4.1(h) and 4.1(i). A particularity of this input image is that some of the patterns initially assigned by the clustering algorithm are discarded or changed during classification, which is indicated by the change of gray levels from the pattern image, Figure 4.1(b), to the last classified image, Figure 4.1(h).

Although, in terms of functionality, the proposed technique resembles the approaches in (Ojala and Pietikäinen, 1999; Camilleri and Petrou, 2000), discussed in Section 2.4.4, it is conceptually different. While the latter applies a supervised classifier in order to refine the boundaries of a complete segmentation produced by a previous unsupervised stage, the methodology proposed in this thesis aims at completely classifying the input image, using the patterns obtained by an approximate unsupervised segmentation of the same image. In other words, while the main task in those previous approaches is performed through unsupervised clustering, the main task in the present approach is carried out by supervised classification. Therefore, clustering is only a means for obtaining training patterns and it may be replaced for any suitable, alternative method.

On the other hand, in contrast to (Mirmehdi and Petrou, 2000; Luo and Savakis, 2000; Kim and Kang, 2007), which have also been discussed in Section 2.4.4, the proposed technique considers information in the image domain instead of information in the feature space when selecting the samples corresponding to the training models. In other words, while the training feature vectors in those previous approaches are directly selected from clusters (closest pixels to the centroid or the centroid itself), in our approach, pixels are selected from the segmented image obtained after clustering (inner region pixels), which improves the generalization of the classifier used for segmentation, as more diverse instances are learnt. Although some outlier instances may

also be learnt by following this strategy, most of them are removed while descending along the levels of the proposed top-down multi-window scheme. This is another key point of the proposed algorithm, as it allows the training texture models to be updated and improved after each classification level. Experimental results shown in Section 4.3.4 demonstrate the advantages of the proposed technique compared, for instance, to the refinement method by Luo and Savakis (2000), which is a good representative of the above approaches and the most similar to ours.

The main aspects of the proposed technique are detailed in the remaining of this section.

4.2.1 Pattern Discovery

In this stage, a subset of the previously obtained feature vectors is passed to a clustering algorithm in order to discover the texture classes present in the processed image. Since clustering is a usual approach related to many applications in different domains, there are several alternatives regarding the clustering algorithm to be used (refer to Jain et al., 2000; Omran et al., 2007, for surveys). In this thesis, three baseline algorithms that are thought to be representative of the existing clustering approaches have been considered for the core of the pattern discovery stage: k-means, mean shift clustering and graph clustering based on the normalized cut. The following sections describe modifications of these baseline algorithms in order to overcome some of their limitations, such as determining an appropriate partition and reducing the computational burden.

4.2.1.1 K-Means-Based Clustering

The iterative k-means algorithm, as well as its strengths and weaknesses, have already been discussed in Section 3.3.1.1. in the context of prototype computation. As a solution for the main drawbacks of k-means, a resolution-driven clustering (RDC)

algorithm has been proposed. It has been shown that this algorithm has many advantages over the classical k-means and, thus, it is a good candidate for the pattern discovery stage.

Since the application of clustering is more critical in the unsupervised case than in the supervised one, as there are no labels to guide the clustering process and not only representatives (prototypes) of the known patterns must be determined, but the patterns themselves, a modified version of k-means by following the methodology proposed for the RDC algorithm has been utilized. The resulting algorithm proceeds as follows:

1. *Split*: suppose that the data points have already been split into C disjoint clusters (initially $C = 1$). Then, the *intra-cluster mean distance*, i.e., the mean distance between the centroid and its associated points, is computed for each cluster and the *global mean distance*, defined as the mean of the intra-cluster mean distances, is obtained for the whole partition. If this global mean exceeds a threshold, the largest cluster in terms of intra-cluster mean distance is split into two ($C = C + 1$). The splitting is done by finding the main principal component ϱ of the cluster and initializing two new child centroids at $\varphi \pm \delta$, where φ is the centroid of the cluster to be split and $\delta = \varphi\sqrt{2\lambda/\pi}$, with λ being the eigenvalue associated with the main principal component ϱ.

2. *Refinement*: after splitting, the refinement stage consists of applying k-means using the C available centroids as initial seeds. Both split and refinement are iterated until no new clusters are generated.

As with the RDC algorithm, the above modified k-means has two main advantages with respect to the classical version. First, instead of the desired number of clusters, the global mean distance threshold controls the output of the algorithm. Such a threshold parameter is more intuitive and more closely related to the perceptual properties of an image (encoded as feature vectors) than a number of clusters. Second,

due to its seed placement method, the algorithm is deterministic.

4.2.1.2 Mean Shift-Based Clustering

The clustering algorithm based on mean shift proposed by Comaniciu and Meer (2002) has also been discussed in Section 3.3.1.1. Unfortunately, it becomes impractical in the context of texture segmentation due to the expensive computation required in order to find the nearest neighbors of a point in a high-dimensional space. Although some approximate strategies for solving this problem have been studied (e.g., Pan et al., 1996; Georgescu et al., 2003), they are not efficient enough (for instance, the experiments in Section 5.3 evaluate the results by Georgescu et al., 2003). Thus, in this thesis, an alternative approximate version has been utilized. It consists of the following steps:

1. Initialize the mean shift procedure on a given point and iterate as usual until a stationary point (mode candidate) is reached.

2. At each iteration, mark all points involved in the mean shift computation as "already visited". They are not taken as initial points anymore.

3. Assign a vote to those visited points regarding their membership to the cluster associated with the detected mode.

4. Repeat steps 1 to 3 with the remaining "not visited" points.

Once all mode candidates have been found, mode merging is performed by means of the same approximate mean shift algorithm, by considering the found modes as data points. If two modes are merged, their membership votes are merged as well, thus keeping track of the new structure of clusters. The mode merging step is repeated until no modes can be merged. The membership of each point is finally determined by majority voting. Although this approximate strategy does not perform an exhaustive analysis of the feature space (as it is usually done) and may lead to a certain loss

UNIVERSITAT ROVIRA I VIRGILI
SUPERVISED AND UNSUPERVISED SEGMENTATION OF TEXTURED IMAGES BY EFFICIENT MULTI-LEVEL PATTERN CLASSIFICATION
Jaime Christian Meléndez Rodríguez
ISBN:978-84-693-7671-3/DL:T-1750-2010

of accuracy, a highly accurate segmentation at this stage is not necessary for the complete segmentation technique to be successful.

Some implementation details are given next. First, the selection of points for initialization of the mean shift procedure can be done randomly or according to some predefined order, which obviously decides if non-deterministic or deterministic clustering is achieved. Since deterministic outputs are preferred, the evaluation order has always been the same and corresponds to the one followed during feature extraction. Second, the bandwidth used for both mode detection and mode merging is the same. Third, although Comaniciu and Meer (2002) suggest that results obtained with the normal kernel are typically better than those obtained with the uniform kernel, the latter has been selected, since its convergence is faster and, as stated earlier, a high quality initial segmentation is not mandatory.

4.2.1.3 Graph Clustering Based on the Normalized Cut

As discussed in Section 3.3.1.2, the graph clustering algorithm based on the normalized cut proposed by Shi and Malik (2000) has become popular in the image segmentation domain. However, the main drawback of this approach is that the computational technique for minimizing the normalized cut is based on eigenvectors. Therefore, it suffers from scalability problems, since in cases where the number of data points is very large, eigenvector computation becomes prohibitive.

Recently, Dhillon et al. (2007) demonstrated the mathematical equivalence between the objective functions used by weighted graph clustering methods and a generalization of the standard k-means algorithm known as *weighted kernel k-means*. Based on this equivalence, they have developed a more efficient technique referred to as GRACLUS, which embeds the weighted kernel k-means algorithm into a multilevel approach in order to locally optimize a number of graph clustering objectives, among them, the normalized cut.

However, before applying GRACLUS to the pattern discovery stage, the prob-

lem of specifying the number of clusters must be addressed such as with k-means. Usually, the alternative is to first bipartition the whole graph and then repartition the already segmented parts if the normalized cut is below a specified value (Shi and Malik, 2000). Notwithstanding, a slightly different approach that follows the ideas of the modified k-means described in Section 4.2.1.1 has been applied. Therefore, the proposed algorithm first partitions the whole graph into two clusters and, if the value of the normalized cut does not reach a specified threshold, it increases the number of clusters by one and repartitions the whole graph using this new setting. This process is repeated until the desired threshold is satisfied.

Preliminary experimentation indicates that this approach finds out the texture patterns present in the analyzed image more accurately than the one proposed in (Shi and Malik, 2000). In addition, it establishes a relationship between a threshold and the resulting clusters as the modified k-means of Section 4.2.1.1 does. As a result, it has the same advantage related to intuitiveness and perception with respect to the original algorithm in (Dhillon et al., 2007).

4.2.2 Supervised Pixel-Based Classification

At this stage, the set of texture patterns found by the previous pattern discovery stage are used as training texture models for a supervised pixel-based classifier, thus effectively transforming the original unsupervised problem into a supervised one.

As stated above, the idea is to utilize the methodology proposed in Chapter 3, which has yielded three pixel-based classifiers referred to as GMM-RDC-KNN, GMM-NCC-KNN and GMM-SVM-OAO, along with its corresponding efficiency-improved versions. However, since the proposed prototype reduction and feature selection strategies (Sections 3.4.1 and 3.4.3, respectively) require additional classification procedures involving the training texture models besides the classification of the test image, they are not feasible in the present context, as both training and classification are now carried out online and, thus, both account for the global processing time. In

this sense, those strategies will rend the segmentation process slower instead of faster, as training with them is more expensive than classification.

On the contrary, the support vector reduction method of Section 3.4.2 is very simple and only consists of a sum (equation 3.17). Furthermore, this method yielded the fastest classifier among the developed ones and its classification rate was very close to the best. In consequence, GMM-SVM-OAO+SVR is the chosen alternative for the pixel classification stage of the proposed technique.

4.3 Experimentation

This section illustrates the application of the proposed unsupervised texture segmentation technique. It describes the experimental setup, as well as how the segmentation technique is configured for the experiments, and shows and discusses the obtained results. Three sets of experiments are carried out. In the first set of experiments, the benefits of including the supervised pixel-based classification stage are assessed by comparing the initial segmentations obtained by the pattern discovery stage with those obtained by the complete technique. In the second set of experiments, the proposed technique is compared with two-stage approaches resulting from combining the original versions of the clustering algorithms considered for pattern discovery with the supervised refinement method proposed in (Luo and Savakis, 2000). Finally, in the last set of experiments, the robustness of the proposed technique is evaluated.

4.3.1 Experimental Setup

The proposed technique has been evaluated on a wide variety of synthetic and real images corresponding to all the sets given in Appendix A (Figures A.1, A.2, A.4, A.6 and A.8). The same remark as in Section 3.5.1 regarding the subjective boundaries associated with the real scenes (sets c to e) is also valid in the unsupervised case.

In addition, images may have more than one possible ground-truth, as the good-

UNIVERSITAT ROVIRA I VIRGILI
SUPERVISED AND UNSUPERVISED SEGMENTATION OF TEXTURED IMAGES BY EFFICIENT MULTI-LEVEL PATTERN CLASSIFICATION
Jaime Christian Meléndez Rodríguez
ISBN:978-84-693-7671-3/DL:T-1750-2010

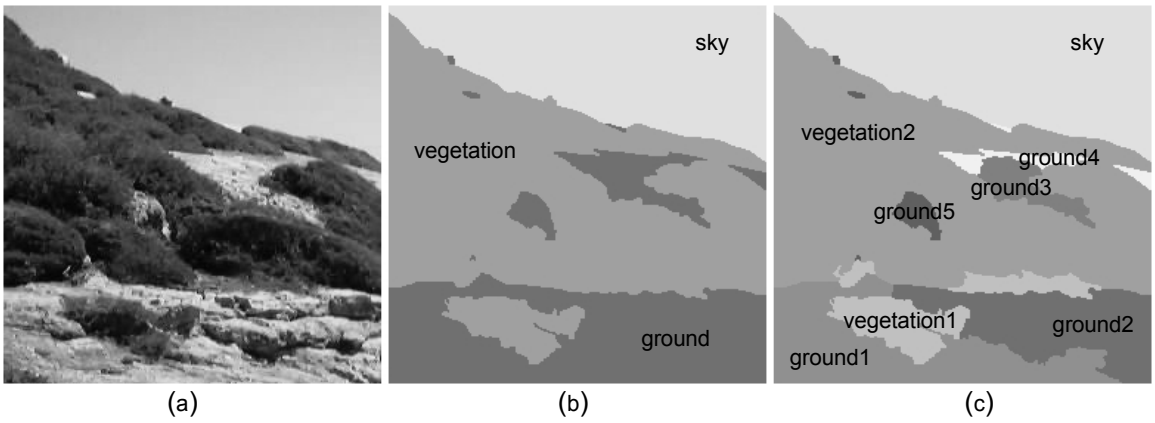

Figure 4.2: Examples of possible segmentations for a real scene. Different classes are represented with different gray levels. Images are magnified in order to appreciate the details that allow the distinction between coarse (b) and fine (c) segmentation.

ness of segmentation depends on the desired level of detail, which, in turn, depends on the context of the application for which the segmentation technique is required. This is especially true for some real scenes, such as those depicted in images 15 to 19 and 22 to 25, in which the number of different textured regions is not clear. For instance, Figure 4.2 shows such a scene, Figure 4.2(a), and two possible segmentations (ground-truths). While the first segmentation, Figure 4.2(b), tries to achieve a *coarse* description of the scene by only considering three classes (ground, vegetation and sky), the second segmentation, Figure 4.2(c), seeks for a *fine* description that considers eight classes, by including different types of ground and vegetation. On the contrary, this problem does not apply to the synthetic compositions, since they are constituted by patches that belong to known texture classes. Therefore, a single ground-truth suffices. Something similar happens with the remaining real scenes. Thus, the results reported in the experiments corresponding to unsupervised segmentation take into account both coarse and fine ground-truths per image.

In order to measure the quality of the segmentation maps produced by the evaluated approaches, a segmentation quality factor Q has been defined. The latter is a more elaborated version of the ideas proposed in (Garcia and Puig, 2003) and is

inspired by the classification rate that measures the performance of supervised pixel-based classifiers, which is simply the ratio between the number of correctly classified pixels and the number of valid pixels in the ground-truth. However, in the unsupervised case, it is not possible to determine which pixels are correctly classified due to the lack of correspondence between the labels of the segmentation map and the labels of the ground-truth. Thus, in order to adapt the above performance measure, a region-based quality factor instead of a pixel-based one has been utilized.

The basic idea consists of comparing the segmentation map with the corresponding ground-truth and, for every region in the latter, determining the region with the largest overlap in the segmentation map and then associating both regions. The ratio between the area of the overlapping portion and the area of the corresponding ground-truth region is an indicator of how good the segmentation of that particular region is. Afterwards, a global score can be obtained by aggregating the partial scores:

$$Q = \sum_{h=1}^{H} \frac{\Delta_h^{gt} \bigcap \Delta_h^{sm}}{\Delta_h^{gt}}, \tag{4.1}$$

where Δ_h^{gt} is the area of one of the H regions in the ground-truth, Δ_h^{sm} is the area of its corresponding region in the evaluated segmentation map and $\Delta_h^{gt} \bigcap \Delta_h^{sm}$ is the area of the overlapping portion between both regions. The area of a region is defined as the number of its pixels. In this way, the key aspect of the analogous pixel-based quality measure is incorporated.

In contrast to (Garcia and Puig, 2003), associations between regions are ensured to be unique. The order in which they occur depends on the number of overlapping pixels and starts with those pairs with the largest values. By following this association methodology, better penalization for both *oversegmentation* and *undersegmentation* is achieved. For instance, when oversegmentation occurs, multiple regions in the segmentation map can be associated with a single region in the ground-truth. However, since the latter is associated with only one region in the segmentation map, the remaining regions are discarded by the pairing process and do not increment Q.

UNIVERSITAT ROVIRA I VIRGILI
SUPERVISED AND UNSUPERVISED SEGMENTATION OF TEXTURED IMAGES BY EFFICIENT MULTI-LEVEL PATTERN CLASSIFICATION
Jaime Christian Meléndez Rodríguez
ISBN:978-84-693-7671-3/DL:T-1750-2010

Alternatively, if undersegmentation occurs, not all regions in the ground-truth can be associated with the regions in the segmentation map. In those cases, the partial score achieved for the regions that cannot be associated is zero. Thus, they do not contribute to Q either.

Since the previous formulation disregards the "relevance" of the analyzed region, a weighting term defined as the ratio between the area of the analyzed region and the area of the whole image has been included. In this way, large regions become more relevant than small ones, as the former contribute with larger numbers of pixels. In consequence, equation 4.1 is reformulated as

$$Q = \sum_{h=1}^{H} \frac{\Delta_h^{gt} \bigcap \Delta_h^{sm}}{\Delta_h^{gt}} \frac{\Delta_h^{gt}}{\Delta_{\mathbf{I}}}, \tag{4.2}$$

which after simplification becomes

$$Q = \frac{1}{\Delta_{\mathbf{I}}} \sum_{h=1}^{H} \Delta_h^{gt} \bigcap \Delta_h^{sm}, \tag{4.3}$$

where $\Delta_{\mathbf{I}}$ is the area of the whole image. It is important to note that, as a result of the evaluation window (or windows) applied during feature extraction, the stripe of unprocessed pixels at the boundary of the segmentation map, as discussed in Section 3.2, is also present in the unsupervised case. These pixels are not taken into account when computing Q. They appear in black in the shown segmentation maps.

The above formulation bounds the score between zero and one, although the $[0, 100]$ interval has been preferred. Figure 4.3 shows examples of different segmentation maps and the scores achieved by applying the proposed quality measure considering the ground-truth in Figure 4.3(a). Figure 4.3(b) corresponds to the perfect case, where all the regions in the segmentation map match those in the ground-truth exactly. Figure 4.3(c) corresponds to a good, although not perfect segmentation, in which region boundaries are wrong. Therefore, its score is close to the maximum. Finally, Figure 4.3(d) corresponds to a bad segmentation in which the left side of the image has been oversegmented and the right side has been undersegmented. Thus,

UNIVERSITAT ROVIRA I VIRGILI
SUPERVISED AND UNSUPERVISED SEGMENTATION OF TEXTURED IMAGES BY EFFICIENT MULTI-LEVEL PATTERN CLASSIFICATION
Jaime Christian Meléndez Rodríguez
ISBN:978-84-693-7671-3/DL:T-1750-2010

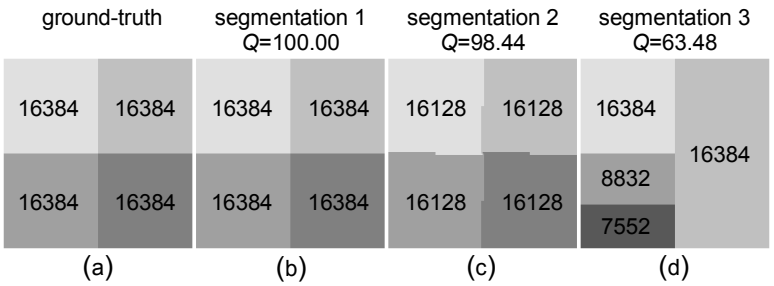

Figure 4.3: Examples of segmentation maps and scores achieved by applying the proposed segmentation quality measure. For the ground-truth (a), the numbers inside the regions indicate their number of pixels. For the segmentation maps (b-d), the numbers inside the regions are the number of overlapping pixels with respect to their associated regions in the ground-truth. Pairs of associated regions have been assigned the same gray level as in the ground-truth. The non-associated region in (d) is indicated with the darkest gray.

the obtained score is low. Note that the bottom right region in the ground-truth has no associated region in the segmentation map, as the only overlapping region has already been associated with the upper right one. Something similar happens with the bottom left region in the segmentation map, which has been discarded by the pairing process in having the least number of overlapping pixels.

Computation times for all experiments in this chapter correspond to an Intel Pentium 4 at 3.2 GHz with 2 GB of RAM. They are always given in seconds.

4.3.2 Segmenter Configuration

The proposed technique is configured as follows. For texture feature extraction, the same Gabor filter bank and texture features as in Section 3.5.2 are utilized. Features are computed for multiple evaluation window sizes as well, although only three window sizes are considered: 5×5, 9×9 and 17×17. The 33×33 and 3×3 windows have been discarded after analyzing and complementing the experimental results of Section 3.5.3. For example, less than 1/3 of the evaluated instances require the 33×33 window and, if the 17×17 window is set as the largest window instead, only a 0.28%

of degradation in classification accuracy is experienced for those instances. On the other hand, less than 10% of pixel classifications is performed by the 3×3 window. Thus, its contribution is minimal.

In addition, the utilization of the various evaluation window sizes differs for each of the stages of the proposed technique. For the pattern discovery stage, a square window of 17×17 pixels is applied, since, according to the above discussion, this is the minimum window size that, in general, allows capturing sufficient texture information in order to appropriately discriminate between the sought patterns. For the supervised classification stage, all the three evaluation window sizes are used.

For the pattern discovery stage, the subset of feature vectors passed to the clustering algorithms corresponds to 1/16 of the total and has been obtained by sampling their associated image pixels using a factor of 1/4 along both the horizontal and vertical directions. This value is a good compromise between reducing the number of feature vectors and appropriately capturing those texture classes with small spatial extent. Since the core of this stage is a clustering algorithm, it is necessary to configure the parameter that controls the search for clusters (classes), which in all cases is a threshold (Sections 4.2.1.1 to 4.2.1.3). This is the only parameter left for user configuration and determines the level of detail achieved in segmentation. Along the experiments, this parameter is set to its optimal value for each test image unless the contrary is stated.

Regarding the pixel-based classification stage, the GMM-SVM-OAO+SVR classifier with the simplest linear kernels is utilized. SVMs are trained with as few as one hundred samples per texture pattern. In general, the smaller the number of training samples, the lower the computation time during training and, especially, during classification, since the chances of obtaining fewer support vectors are higher, although the latter is not a problem due to the applied SV reduction method. On the other hand, if the number of training samples is too low, important information about the patterns may be missed, leading to a significant degradation in classification (segmen-

UNIVERSITAT ROVIRA I VIRGILI
SUPERVISED AND UNSUPERVISED SEGMENTATION OF TEXTURED IMAGES BY EFFICIENT MULTI-LEVEL PATTERN CLASSIFICATION
Jaime Christian Meléndez Rodríguez
ISBN:978-84-693-7671-3/DL:T-1750-2010

tation) accuracy. In this sense, the value of one hundred samples considered here can be thought of as a lower bound for this parameter. The post-processing procedure is the same as the one described in Section 3.5.2.

4.3.3 First Set of Experiments

The goal of the first set of experiments is to assess the benefits of including supervision, as described in Section 4.2.2, in the proposed unsupervised segmentation scheme. To achieve this goal, segmentations produced by the complete scheme have been compared with those produced by the pattern discovery stage of Section 4.2.1 in terms of Q defined in Section 4.3.1 and processing time (which is the sum of feature extraction and segmentation times). The algorithms corresponding to the pattern discovery stage are denoted as GMM-PD$_{\mathrm{KMC}}$ (Section 4.2.1.1), GMM-PD$_{\mathrm{MSC}}$ (Section 4.2.1.2) and GMM-PD$_{\mathrm{GC}}$ (Section 4.2.1.3), whereas the complete scheme (including pixel-based classification) associated with each of them is denoted as GMM-PD$_{\mathrm{KMC}}$-PBC, GMM-PD$_{\mathrm{MSC}}$-PBC and GMM-PD$_{\mathrm{GC}}$-PBC, respectively.

The average results for the first set of experiments are summarized in Table 4.1. Figures 4.4 and 4.5 show some segmentation maps produced by the evaluated approaches. The results indicate that the pixel-based classification stage significantly improves the segmentation quality achieved by the preliminary segmentation of the pattern discovery stage, which is more evident in those cases where GMM-PD$_{\mathrm{KMC}}$ and GMM-PD$_{\mathrm{MSC}}$ are applied. The reason is that both algorithms alone (especially GMM-PD$_{\mathrm{MSC}}$) tend to oversegment the image when discriminating between texture classes is difficult. By including the pixel-based classification stage, texture classes are better identified and most irrelevant/outlier patterns are merged or discarded along the descending classification levels, thus producing a segmentation that better reflects the image region structure.

Although the improvement for GMM-PD$_{\mathrm{GC}}$ is not as dramatic as for the other pattern discovery algorithms, it is still important and most appreciable in complex

UNIVERSITAT ROVIRA I VIRGILI
SUPERVISED AND UNSUPERVISED SEGMENTATION OF TEXTURED IMAGES BY EFFICIENT MULTI-LEVEL PATTERN CLASSIFICATION
Jaime Christian Meléndez Rodríguez
ISBN:978-84-693-7671-3/DL:T-1750-2010

Table 4.1: Average segmentation quality ($\overline{Q}$), average feature extraction time ($\overline{FET}$), average segmentation time ($\overline{ST}$) and average processing time ($\overline{PT}$) for the approaches evaluated in the first set of experiments.

Segmentation technique	$\overline{Q}$ sets a-b	$\overline{Q}$ sets c-e	$\overline{Q}$ all sets	$\overline{FET}+\overline{ST}=\overline{PT}$ all sets
GMM-PD$_{\text{KMC}}$	76.04	74.67	75.03	0.530+0.136=0.666
GMM-PD$_{\text{KMC}}$-PBC	89.82	85.63	86.74	1.332+0.394=1.726
GMM-PD$_{\text{MSC}}$	65.14	76.36	73.39	0.530+0.817=1.347
GMM-PD$_{\text{MSC}}$-PBC	87.22	86.99	87.05	1.332+1.499=2.831
GMM-PD$_{\text{GC}}$	87.00	79.02	81.13	0.530+2.771=3.301
GMM-PD$_{\text{GC}}$-PBC	92.86	88.59	89.72	1.332+2.940=4.272

images, such as the second synthetic composition (column 2 of Figure 4.4), the fine segmentation of outdoor scenes 15 and 25 (columns 4 and 6 of Figure 4.4) and the real images 39, 40, 41 and 45 (columns 2, 3, 4 and 6 of Figure 4.5). In those cases, the top-down approach followed by the classification stage not only performs boundary refinement in the image domain, but mainly pattern refinement in the feature space. For instance, notice how the left patch in image 14, the sea, the vegetation and the cape in image 15, most types of soil in image 25, the cloud in image 39 or the ground patterns in image 45 have been correctly segmented by GMM-PD$_{\text{GC}}$-PBC even when GMM-PD$_{\text{GC}}$ could not do it.

In terms of computation time, the pattern discovery algorithms are the fastest, as expected, since they utilize features evaluated over a single window and only process 1/16 of the feature vectors. However, as discussed above, their segmentation quality is low. On the other hand, the complete segmentation techniques achieve competitive times and considerably better segmentation quality, especially in the case of GMM-PD$_{\text{GC}}$-PBC, where the overhead introduced by the pixel-based classification stage is less than 25% and mainly due to feature extraction.

Among the three versions of the proposed technique, GMM-PD$_{\text{GC}}$-PBC is the

UNIVERSITAT ROVIRA I VIRGILI
SUPERVISED AND UNSUPERVISED SEGMENTATION OF TEXTURED IMAGES BY EFFICIENT MULTI-LEVEL PATTERN CLASSIFICATION
Jaime Christian Meléndez Rodríguez
ISBN:978-84-693-7671-3/DL:T-1750-2010

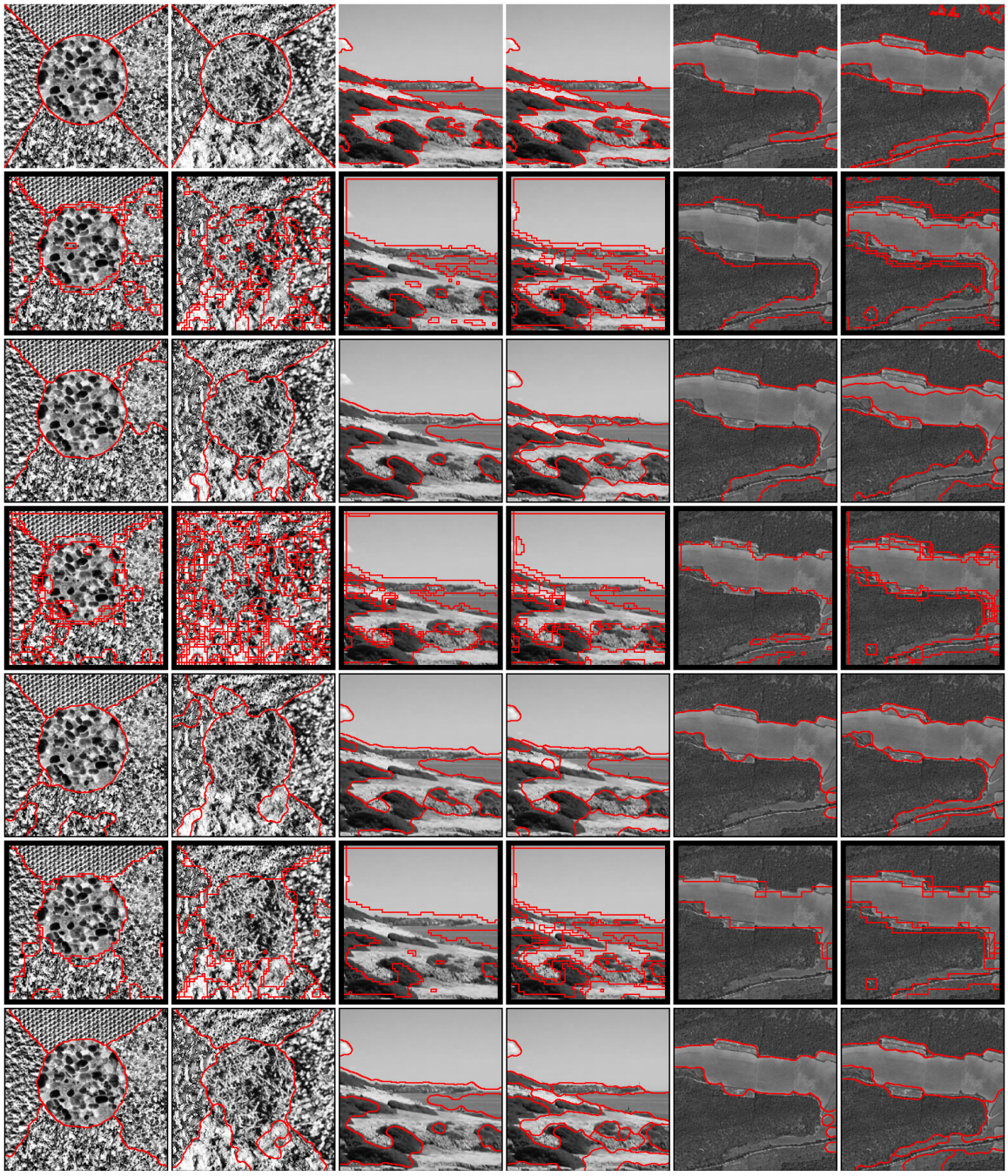

Figure 4.4: Segmentation maps for images 5, 14, 15 and 25. First row: ground-truth. Starting from the second row, results by: GMM-PD$_{\text{KMC}}$, GMM-PD$_{\text{KMC}}$-PBC, GMM-PD$_{\text{MSC}}$, GMM-PD$_{\text{MSC}}$-PBC, GMM-PD$_{\text{GC}}$ and GMM-PD$_{\text{GC}}$-PBC.

UNIVERSITAT ROVIRA I VIRGILI
SUPERVISED AND UNSUPERVISED SEGMENTATION OF TEXTURED IMAGES BY EFFICIENT MULTI-LEVEL PATTERN CLASSIFICATION
Jaime Christian Meléndez Rodríguez
ISBN:978-84-693-7671-3/DL:T-1750-2010

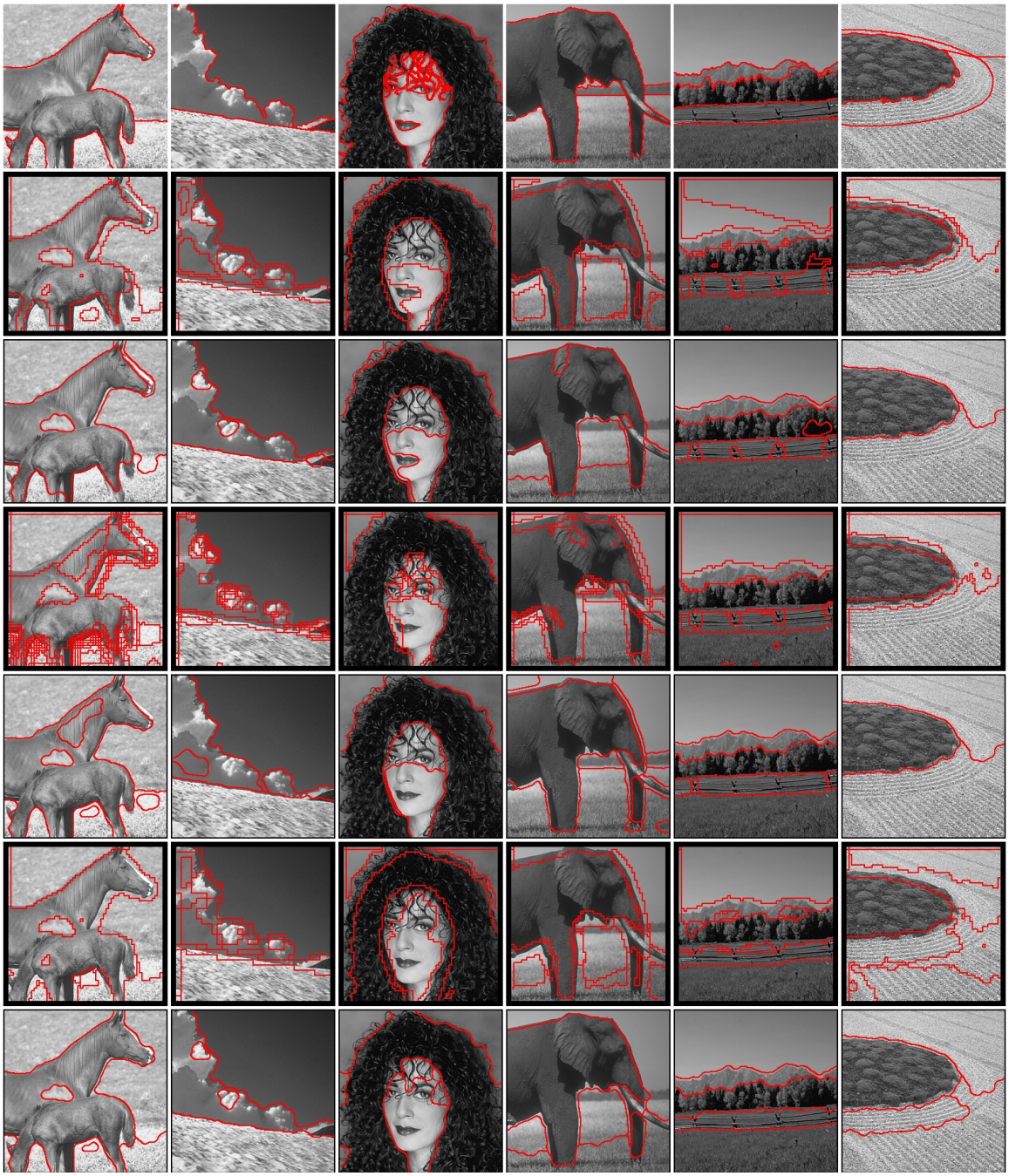

Figure 4.5: Segmentation maps for images 35, 39, 40, 41, 43 and 45. First row: ground-truth. Starting from the second row, results by: GMM-PD_{KMC}, GMM-PD_{KMC}-PBC, GMM-PD_{MSC}, GMM-PD_{MSC}-PBC, GMM-PD_{GC} and GMM-PD_{GC}-PBC.

one that achieves the best segmentation quality, although it is the slowest due to the graph-based clustering algorithm. In this case, most of the segmentation time is spent in computing the dissimilarity matrix, which takes around 2 seconds for 3600 feature vectors of 48 dimensions. On the contrary, GMM-PD$_{\text{KMC}}$-PBC is the fastest approach, since it utilizes the simplest clustering strategy. However, its segmentation quality is the lowest. Finally, GMM-PD$_{\text{MSC}}$-PBC performs in the middle, although it is the version that introduces the highest overhead in the pixel-based classification stage, as this stage has to deal with the large number of classes resulting from the oversegmentation produced by the mean shift procedure.

4.3.4 Second Set of Experiments

In order to provide further insight regarding supervision, in the second set of experiments, the original clustering algorithms reviewed for the pattern discovery stage, namely: k-means, mean shift and GRACLUS, are combined with the supervised refinement method discussed in (Luo and Savakis, 2000), thus yielding a set of two-stage approaches, and are compared with their corresponding versions of the proposed segmentation technique. The resulting two-stage approaches are referred to as GMM-k-means+R, GMM-mean shift+R and GMM-GRACLUS+R. The refinement method by Luo and Savakis (2000) has been chosen because it is a good representative of the ones reviewed in Section 2.4.4 and it is the most similar to the method proposed in this thesis. Comparisons are carried out in the same performance terms as in the previous set of experiments.

The configuration of the two-stage approaches considers optimal clustering parameters for each test image. Since neither the original clustering algorithms or the refinement method are designed to work with multiple window sizes, the single size that yields the best results for the whole image database is used. Contrarily to the modified algorithms utilized for pattern discovery, the two-stage approaches take all the available feature vectors as input. The only exception is GRACLUS, due to the

Table 4.2: Average segmentation quality ($\overline{Q}$), average feature extraction time ($\overline{FET}$), average segmentation time ($\overline{ST}$) and average processing time ($\overline{PT}$) for the approaches evaluated in the second set of experiments.

Segmentation	$\overline{Q}$			$\overline{FET}+\overline{ST}=\overline{PT}$
technique	sets a-b	sets c-e	all sets	all sets
GMM-kmeans+R	82.64	79.51	80.34	0.530+1.276=1.806
GMM-PD_{KMC}-PBC	89.82	85.63	86.74	1.332+0.394=1.726
GMM-mean shift+R	76.61	80.70	79.62	0.530+4513.853=4514.383
GMM-PD_{MSC}-PBC	87.22	86.99	87.05	1.332+1.499=2.831
GMM-GRACLUS+R	88.45	81.77	83.53	0.530+49.852=50.382
GMM-PD_{GC}-PBC	92.86	88.59	89.72	1.332+2.940=4.272

lack of resources to process a complete dissimilarity matrix, which in the present context is not sparse. Therefore, only 1/4 of the feature vectors have been taken into account. An alternative would be to enforce sparsity by applying thresholding techniques or by only considering the nearest neighbors of each feature vector. However, both approaches yield degradation in the obtained segmentation, which increases as the number of remaining points or nearest neighbors decreases.

The average results for this set of experiments are summarized in Table 4.2. It can be observed that, although the refinement method in (Luo and Savakis, 2000) improves the original, complete segmentations to a certain extent, it cannot correct errors due to a misleading or imprecise initial segmentation, or may merge some regions erroneously interpreted as a foreign texture formed by two or more abutting regions. This happens even for GRACLUS, which is by far the most accurate among the baseline clustering algorithms. On the other hand, the proposed technique overcomes most of such problems by exploiting supervision based on the information retrieved at the pattern discovery stage. Figure 4.6 shows some examples from Figures 4.4 and 4.5 that demonstrate this behavior, and compares segmentations produced by both the two-stage approaches and their analogous versions of the proposed technique.

UNIVERSITAT ROVIRA I VIRGILI
SUPERVISED AND UNSUPERVISED SEGMENTATION OF TEXTURED IMAGES BY EFFICIENT MULTI-LEVEL PATTERN CLASSIFICATION
Jaime Christian Meléndez Rodríguez
ISBN:978-84-693-7671-3/DL:T-1750-2010

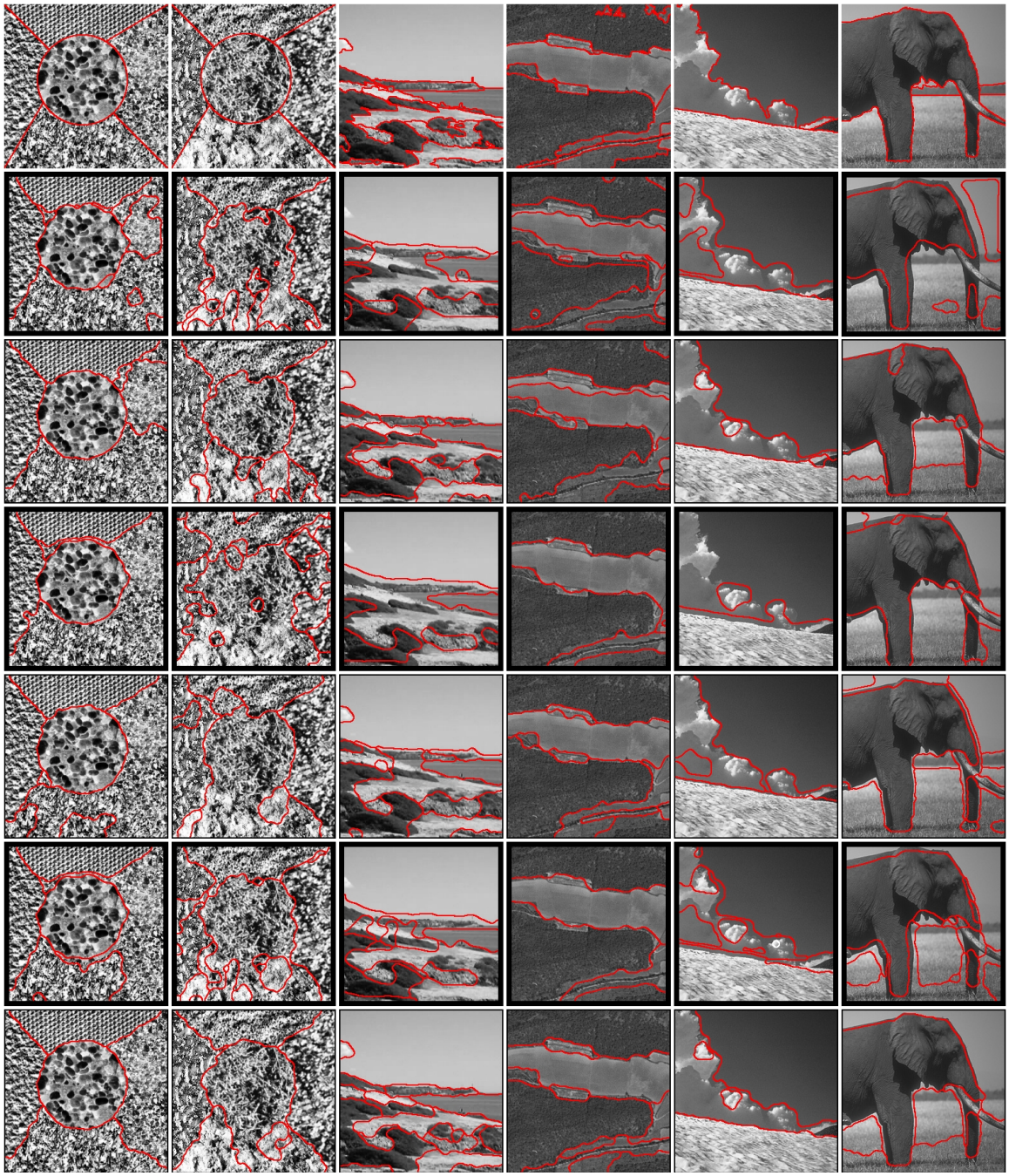

Figure 4.6: Segmentation maps for images 5, 14, 15, 25, 39 and 41. First row: ground-truth. Starting from the second row, results by: GMM-kmeans+R, GMM-PD_{KMC}-PBC, GMM-mean shift+R, GMM-PD_{MSC}-PBC, GMM-GRACLUS+R and GMM-PD_{GC}-PBC.

In terms of computation time, all versions of the proposed technique are faster than their two-stage counterparts, especially in those cases based on mean shift and graph clustering, which is mainly due to the reduced number of feature vectors utilized for pattern discovery, as well as the modification of the original clustering algorithms and the efficient pixel-based classification scheme.

4.3.5 Third Set of Experiments

A desirable property of an unsupervised segmentation technique is *robustness*, that is, the capacity to keep good segmentation quality in the presence of changes in the analyzed scene without parameter reconfiguration. The third set of experiments aims at determining how robust the various versions of the proposed technique are by evaluating the quality of the segmentations produced for a sequence of images using the same parameter configuration. Therefore, for this set of experiments, the parameter that controls the level of segmentation detail of the evaluated approaches has been fixed for the whole sequence. The remaining parameters are kept the same as in the previous experiments.

The sequence of images used in these experiments corresponds to images 15 to 18 in Appendix A. All instances belong to the same scenario and have been acquired under the same conditions. Image 15 may be considered as the reference scene. Image 16 reproduces a simple shift of the camera to the bottom left. Image 17 was produced by forwarding and lowering the camera. Image 18 corresponds to a backward movement and right rotation of the camera. All of the above situations represent typical movements performed during robot exploration. As discussed in Section 4.3.1, both coarse and fine ground-truths are available for each image.

Table 4.3 shows the segmentation quality scores achieved by the three alternatives of the proposed technique. Figures 4.7 and 4.8 show the segmentation maps produced for coarse and fine segmentations, respectively. In general, the three versions of the proposed technique perform well and have only some problems with the fine

UNIVERSITAT ROVIRA I VIRGILI
SUPERVISED AND UNSUPERVISED SEGMENTATION OF TEXTURED IMAGES BY EFFICIENT MULTI-LEVEL PATTERN CLASSIFICATION
Jaime Christian Meléndez Rodríguez
ISBN:978-84-693-7671-3/DL:T-1750-2010

Table 4.3: Segmentation quality (Q) for the approaches evaluated in the third set of experiments.

Segmentation technique	Type of segmentation	Q per image				$\overline{Q}$
		15	16	17	18	
GMM-PD$_{\text{KMC}}$-PBC	coarse	86.52	85.93	84.92	83.02	85.10
	fine	80.72	80.15	79.81	77.08	79.44
GMM-PD$_{\text{MSC}}$-PBC	coarse	82.70	80.94	89.64	85.86	84.79
	fine	80.62	78.63	88.04	71.41	79.68
GMM-PD$_{\text{GC}}$-PBC	coarse	86.51	86.89	87.85	83.51	86.19
	fine	86.28	82.55	85.33	74.89	82.26

segmentation of the last image. A priori, both images 17 and 18 could be thought to be of the same complexity, since they depict similar parts of the scene. Thus, if high scores are obtained for image 17, similar scores are expected for image 18. However, image 18 is more complex, as it introduces new classes, such as the sea and the vegetation in the middle and at the bottom of the scene, as well as more variability of the ground and intensity of the vegetation at the upper half part. Therefore, the low scores achieved in this case are justified.

Among the evaluated approaches, the mean shift-based one is the least robust, as it alternates low and high scores for most of the images. In fact, it achieves the lowest and highest individual scores. On the other hand, the graph clustering-based approach is the most robust and the one that achieves the highest average score for both types of segmentation.

4.4 Summary

This chapter describes a new unsupervised texture segmentation technique that aims at incorporating the advantages of supervision by extending to the unsupervised domain the methodology for supervised texture segmentation developed in the previous

UNIVERSITAT ROVIRA I VIRGILI
SUPERVISED AND UNSUPERVISED SEGMENTATION OF TEXTURED IMAGES BY EFFICIENT MULTI-LEVEL PATTERN CLASSIFICATION
Jaime Christian Meléndez Rodríguez
ISBN:978-84-693-7671-3/DL:T-1750-2010

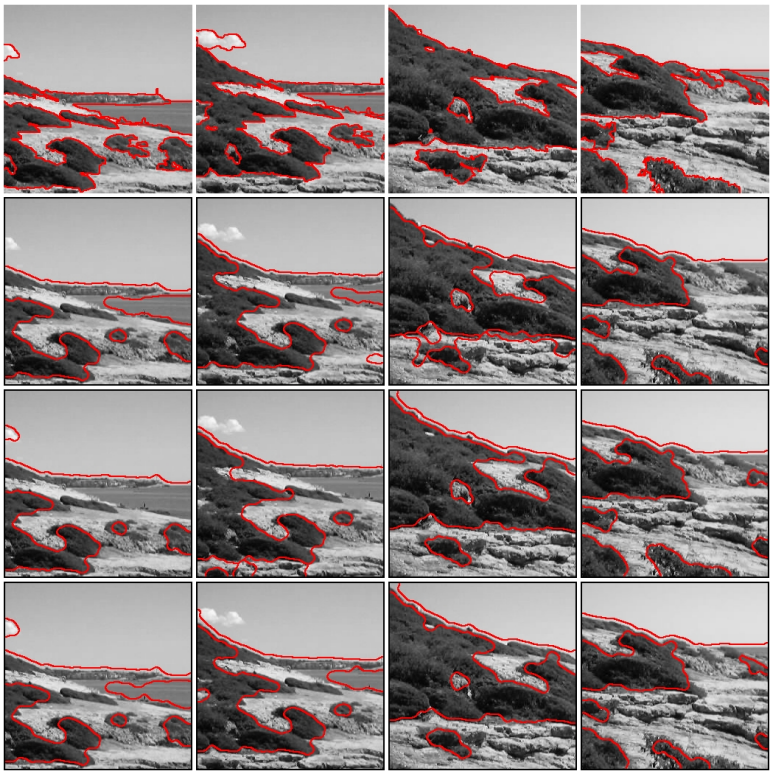

Figure 4.7: Segmentation maps for images 15 to 18 (coarse segmentation). First row: ground-truth. Starting from the second row, results by: GMM-PD$_{\text{KMC}}$-PBC, GMM-PD$_{\text{MSC}}$-PBC and GMM-PD$_{\text{GC}}$-PBC.

chapter.

The proposed technique consists of an initial pattern discovery stage that relies on a clustering algorithm for determining the texture patterns present in a given image. In the second stage, a supervised pixel-based classifier trained with those patterns is used to classify every image pixel into one of the sought texture classes, thus yielding the final segmentation. In this way, the original unsupervised problem is effectively transformed into a supervised one.

Moreover, a segmentation quality measure derived from the classification rate used for evaluating supervised classifiers is also developed. It is based on a pairing process that uniquely associates regions in the ground-truth with regions in the evaluated

UNIVERSITAT ROVIRA I VIRGILI
SUPERVISED AND UNSUPERVISED SEGMENTATION OF TEXTURED IMAGES BY EFFICIENT MULTI-LEVEL PATTERN CLASSIFICATION
Jaime Christian Meléndez Rodríguez
ISBN:978-84-693-7671-3/DL:T-1750-2010

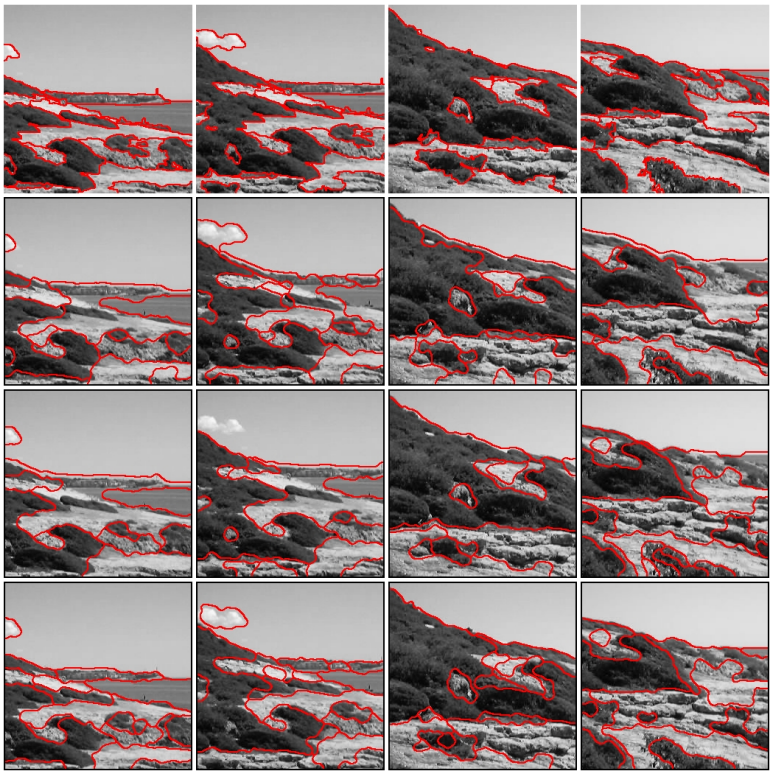

Figure 4.8: Segmentation maps for images 15 to 18 (fine segmentation). First row: ground-truth. Starting from the second row, results by: GMM-PD_{KMC}-PBC, GMM-PD_{MSC}-PBC and GMM-PD_{GC}-PBC.

segmentation map and compares the overlapping between them in order to assess the goodness of a given segmentation. It is shown that this measure adequately penalizes both oversegmentation and undersegmentation.

The conducted experiments indicate that the pixel-based classification stage substantially improves the segmentation quality of the completely unsupervised approaches considered in the pattern discovery stage, with only a moderate increment in processing time. In particular, the accuracy inside regions of homogeneous texture, as well as near boundaries between regions of different texture, is improved by utilizing multiple evaluation window sizes according to the top-down classification scheme developed in the previous chapter. This scheme also contributes to speeding

up the segmentation process, since only a reduced number of image pixels need to be classified by all the considered window sizes. The computational burden is further reduced by utilizing the fastest available classifier, which is based on SVMs.

Experiments also show that, compared to other two-stage approaches, the proposed technique yields the best segmentation quality, as it is able to deal with situations where the initial segmentation is noticeably wrong and, for which, the compared supervised refinement method does not suffice. Moreover, due to its efficient design, the processing time of the proposed technique is lower than the one corresponding to those alternative two-stage approaches.

Finally, the robustness of the three versions of the proposed technique has been assessed. Results indicate that the graph clustering-based version is the most robust. This version is also the one that achieves the highest segmentation quality.

UNIVERSITAT ROVIRA I VIRGILI
SUPERVISED AND UNSUPERVISED SEGMENTATION OF TEXTURED IMAGES BY EFFICIENT MULTI-LEVEL PATTERN CLASSIFICATION
Jaime Christian Meléndez Rodríguez
ISBN:978-84-693-7671-3/DL:T-1750-2010

Chapter 5

Experimental Validation

This chapter presents three sets of experimental results that complement those presented in the previous chapters in order to illustrate and validate the capabilities of the texture analysis methodologies proposed in this thesis. Comparisons with different well-known supervised texture classifiers and unsupervised texture segmenters are provided in order to achieve a more extensive validation. The chapter is organized as follows. Section 5.1 introduces the different sets of experiments that are carried out along the rest of the chapter. Section 5.2 shows the experiments performed on supervised texture segmentation. Section 5.3 describes the results obtained on unsupervised texture segmentation. Section 5.4 compares the results obtained by both the supervised and unsupervised segmentation techniques proposed in this dissertation. The contents of the chapter are finally summarized in Section 5.5.

5.1 Introduction

The goal of this chapter is to carry out an extensive number of comparisons in order to validate the two texture analysis methodologies proposed in this thesis: on the one hand, the supervised texture segmentation methodology presented in Chapter 3, which is based on a top-down, multi-window, pixel-based classification scheme and,

on the other hand, the unsupervised texture segmentation methodology defined in Chapter 4, which is an extension of the scheme defined in Chapter 3.

For the supervised case, among the three proposed pixel-based classifiers, GMM-RDC-KNN+PR is chosen for the comparisons in this chapter, since it is the technique that achieves the highest classification rate (although by a very small difference) after prototype reduction and without feature selection. Feature selection is not taken into account, as none of the alternative techniques evaluated in these experiments implements this stage. The alternative supervised classifiers considered in these experiments are: the non-parametric classifier proposed in (Melendez et al., 2008), the texture classifier based on integration of multiple methods and windows described in (Puig and Garcia, 2006), The SVM-based approach proposed in (Kim et al., 2002), an extension to pixel-based classification of the LBP classifier proposed in (Ojala et al., 2002) and an extension to pixel-based classification of the MeasTex suite presented in (Smith and Burns, 1997).

For the unsupervised case, among the three proposed techniques, GMM-PD$_{\mathrm{GC}}$-PBC has been chosen, since it is the most robust and the one that yields the best segmentation quality, although its processing time is the largest. The following techniques are considered as alternatives for the comparison: the texture and boundary encoding segmenter proposed in (Rao et al., 2009), the oriented watershed transform and ultrametric contour map-based approach described in (Arbeláez et al., 2009), the compression-based texture merging algorithm proposed in (Yang et al., 2008), the spatial/spectral segmenter developed in (Muneeswaran et al., 2006), locally-sensitive hashing-based adaptive mean shift (Georgescu et al., 2003) and EdgeFlow (Ma and Manjunath, 2000).

Finally, in the last set of experiments, GMM-SVM-OAO+SVR and GMM-PD$_{\mathrm{GC}}$-PBC are utilized, since the classification stage of both techniques is based on the same SVM method.

5.2 Experiments on Supervised Texture Segmentation

The goal of the first set of experiments is to validate the proposed methodology for supervised texture segmentation by performing a comparison with other supervised texture segmentation techniques in terms of classification rates and processing time. The experimental setup for this set of experiments corresponds to the one described in Section 3.5.1. The proposed technique, referred to as GMM-RDC-KNN+PR, is configured as in Section 3.5.2. Details about the alternative evaluated techniques are given below:

1. The non-parametric distance-based classifier proposed in (Melendez et al., 2008) is a preliminary version of GMM-RDC-KNN. Therefore, it utilizes as feature extractors the same optimized Gabor filters discussed in Section 3.5.2, although evaluated over six window sizes (an additional 65×65 window is considered), the same RDC algorithm for prototype computation, the KNN rule for pixel level classification, and a similar post-processing procedure. The main difference with the current GMM-RDC-KNN classifier is the multi-window strategy it follows, already discussed in Sections 2.2 and 3.2, which also modifies its corresponding parameter selection algorithm. This classifier is referred to as GMM-RDC-KNN$'$.

2. The texture classifier based on integration of multiple methods and windows described in (Puig and Garcia, 2006) uses the same features and window sizes as GMM-RDC-KNN$'$, as well as the same post-processing procedure. Its multi-window strategy has also been discussed in Sections 2.2 and 3.2. This classifier is referred to as GMM-IMMW.

3. The SVM-based approach proposed in (Kim et al., 2002) utilizes SVMs as feature extractors and primary classifiers by being directly fed with the raw pixels

of the analyzed image. The SVMs have been trained with one thousand samples per texture pattern and have been independently evaluated over the five window sizes suggested by the authors: 5×5, 9×9, 13×13, 17×17 and 21×21. The final classification is performed by a two-layer neural network. Median filtering is included as post-processing. This classifier is referred to as SVM-NN.

4. The extension to pixel-based classification of the LBP-based classifier originally proposed in (Ojala et al., 2002) utilizes the three available LBP operators: $LBP^{riu2}_{8,1}$, $LBP^{riu2}_{16,2}$ and $LBP^{riu2}_{24,3}$, as well as the combination of them, as feature extractors, and the LBP histograms obtained from samples of 16×16 pixels, as indicated in (Ojala et al., 2002), as texture features. The G (log-likelihood) statistic between texture samples and models is used as a dissimilarity measure. Fifty models per texture pattern have been considered in order to be comparable with the number of prototypes utilized by GMM-RDC-KNN+PR. Since there is not a specific post-processing procedure for this classifier, the one described in Section 3.5.2 is applied. This classifier is referred to as LBP-G.

5. The extension to pixel-based classification of the MeasTex suite originally presented in (Smith and Burns, 1997) is performed in a similar way as for LBP-G, but processing samples of 32×32 pixels, which is the minimum sample size accepted by MeasTex. Both the KNN ($K = 5$) and multivariate Gaussian classifiers included in the suite are independently evaluated with features derived from classical Gabor filters, the fractal dimension, gray level co-occurrence matrices and Markov random fields. Since there is not a specific post-processing procedure for MeasTex, the one described in Section 3.5.2 is applied. This classifier is referred to as MeasTex.

Average results for each image set considered in this comparison are summarized in Table 5.1. Figures 5.1 to 5.4 show some classification maps produced by the aforementioned approaches sorted for each image set according to their classification

UNIVERSITAT ROVIRA I VIRGILI
SUPERVISED AND UNSUPERVISED SEGMENTATION OF TEXTURED IMAGES BY EFFICIENT MULTI-LEVEL PATTERN CLASSIFICATION
Jaime Christian Meléndez Rodríguez
ISBN:978-84-693-7671-3/DL:T-1750-2010

Table 5.1: Average classification rates ($\overline{CR}$) and average processing time ($\overline{PT}$) for the approaches evaluated in the first set of experiments.

	$\overline{CR}$					$\overline{PT}$
Classifier	set a	set b	set c	set d	all sets	all sets
GMM-RDC-KNN+PR	93.40	87.06	92.82	93.61	91.86	1.719
GMM-RDC-KNN′	90.74	80.99	87.48	90.00	87.29	27.767
GMM-IMMW	84.11	76.69	88.09	85.89	84.05	279.648
SVM-NN	79.41	50.97	—	86.90	72.78	122.834
LBP-G	89.88	87.67	66.00	79.07	78.95	56.851
MeasTex	81.77	64.96	69.37	74.14	72.09	387.786

rate. For those approaches that independently evaluate different sets of features or have free parameters, such as the evaluation window size, only those features or parameters that yield the best average classification rate along the entire set of test images are considered. Thus, the reported results correspond to: a 21×21 evaluation window for SVM-NN, features from the combination of the three available LBP operators for LBP-G, and the KNN classifier with classical Gabor filters for MeasTex. Results for SVM-NN are not reported for set c, since the SVMs cannot be trained in a reasonable time for most of its images.

As shown in Table 5.1, the overall average classification rate obtained by the proposed classifier is superior to the one obtained by any of the alternative approaches. Moreover, by comparing these results with those in Table 3.7, it is shown that the other two classifiers developed in Chapter 3 are also more accurate. Notwithstanding, there are certain cases, such as set b, for which LBP-G proves to be a better alternative. In particular, there are two instances, images 23 and 25 (columns 5 and 6 of Figure 5.2), for which the difference in favor of LBP-G is quite noticeable due to the poor capabilities of the Gabor features for distinguishing between certain patches (the left, middle and bottom patches in image 23, and the top and middle patches in image 25) and associating them with their respective texture patterns. In fact, all

UNIVERSITAT ROVIRA I VIRGILI
SUPERVISED AND UNSUPERVISED SEGMENTATION OF TEXTURED IMAGES BY EFFICIENT MULTI-LEVEL PATTERN CLASSIFICATION
Jaime Christian Meléndez Rodríguez
ISBN:978-84-693-7671-3/DL:T-1750-2010

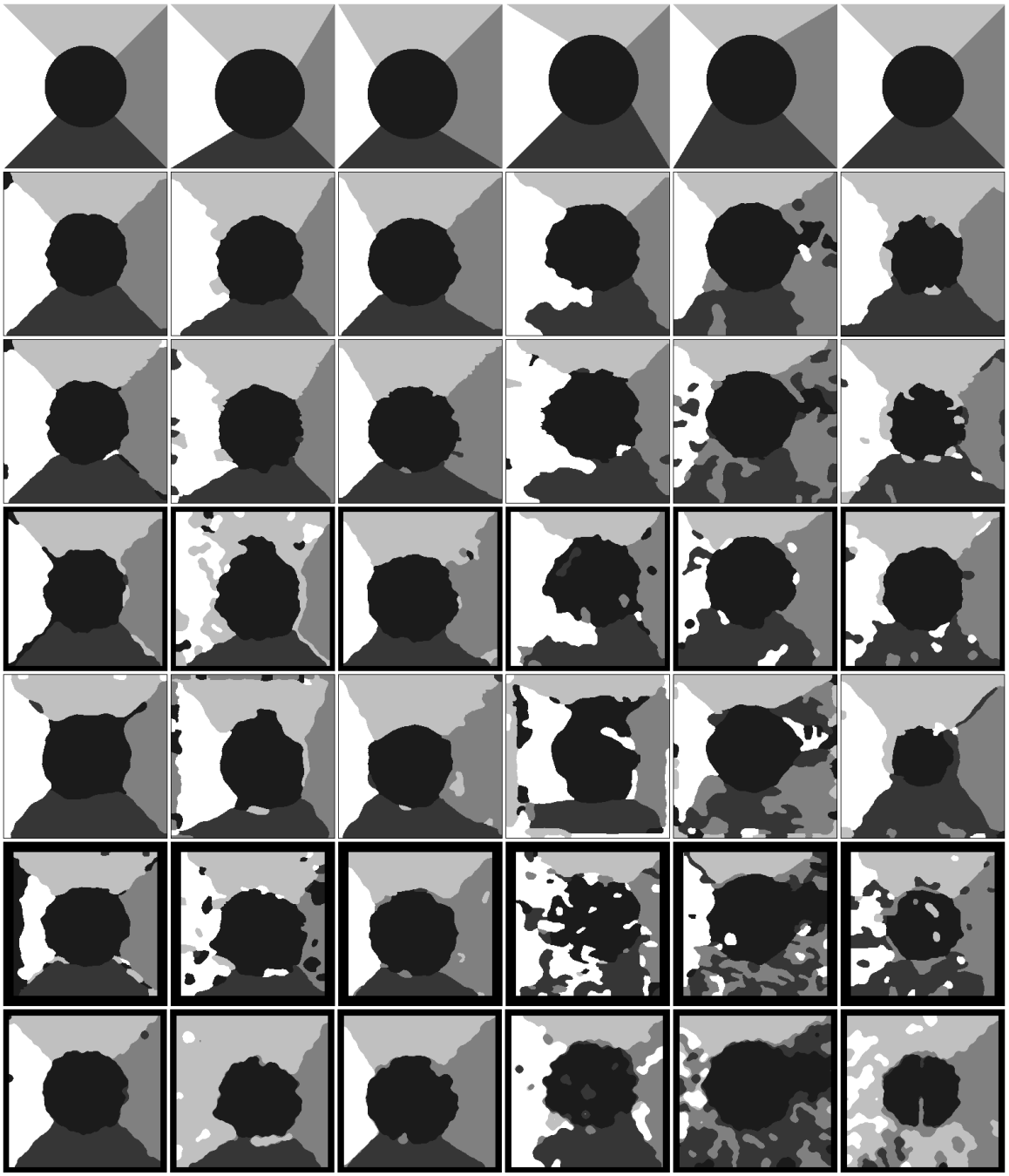

Figure 5.1: Classification maps for images 1 to 6. First row: ground-truth. Starting from the second row, results by: GMM-RDC-KNN+PR, GMM-RDC-KNN′, LBP-G, GMM-IMMW, MeasTex and SVM-NN.

UNIVERSITAT ROVIRA I VIRGILI
SUPERVISED AND UNSUPERVISED SEGMENTATION OF TEXTURED IMAGES BY EFFICIENT MULTI-LEVEL PATTERN CLASSIFICATION
Jaime Christian Meléndez Rodríguez
ISBN:978-84-693-7671-3/DL:T-1750-2010

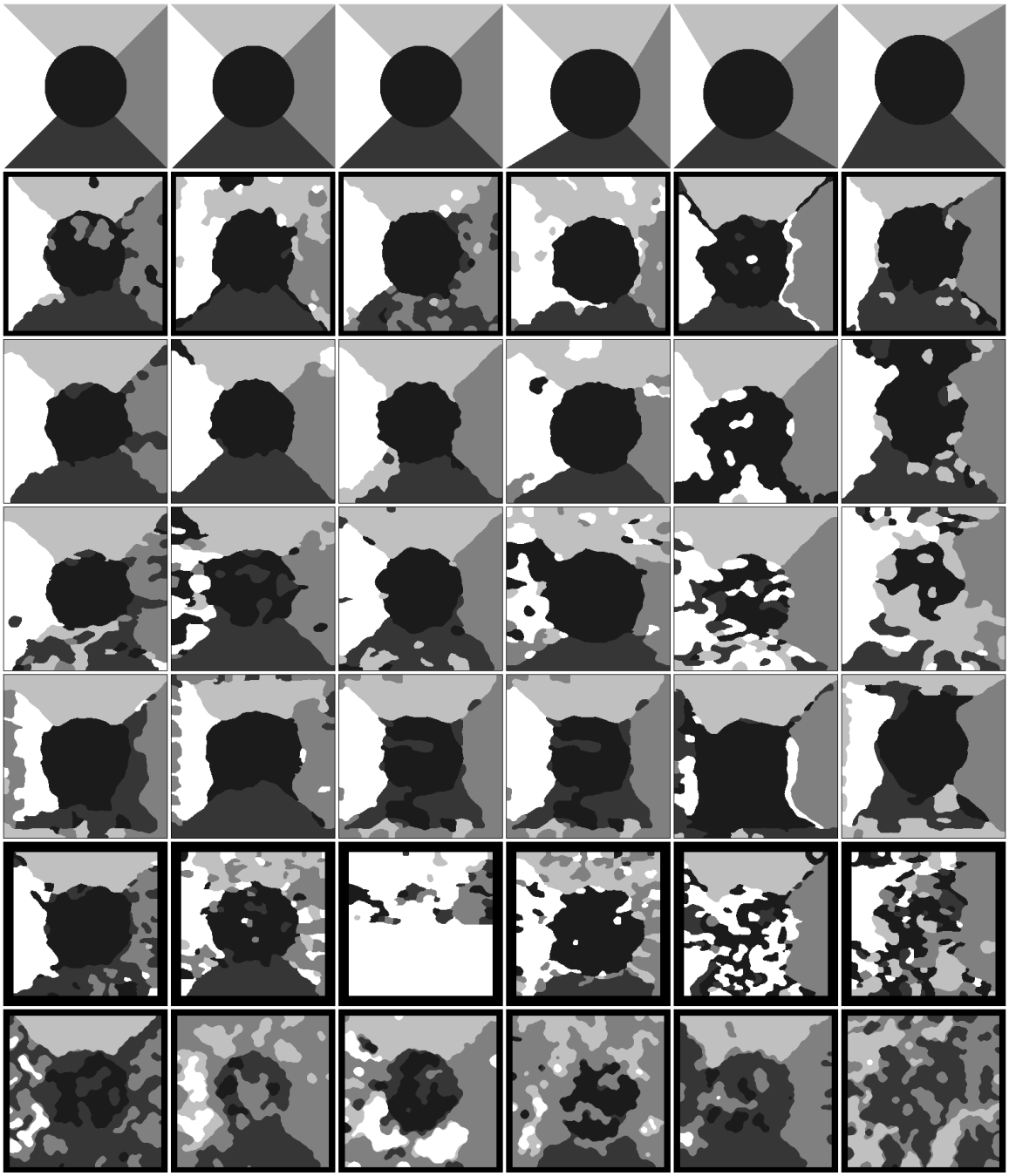

Figure 5.2: Classification maps for images 8, 9, 10, 11, 12 and 14. First row: ground-truth. Starting from the second row, results by: LBP-G, GMM-RDC-KNN+PR, GMM-RDC-KNN′, GMM-IMMW, MeasTex and SVM-NN.

UNIVERSITAT ROVIRA I VIRGILI
SUPERVISED AND UNSUPERVISED SEGMENTATION OF TEXTURED IMAGES BY EFFICIENT MULTI-LEVEL PATTERN CLASSIFICATION
Jaime Christian Meléndez Rodríguez
ISBN:978-84-693-7671-3/DL:T-1750-2010

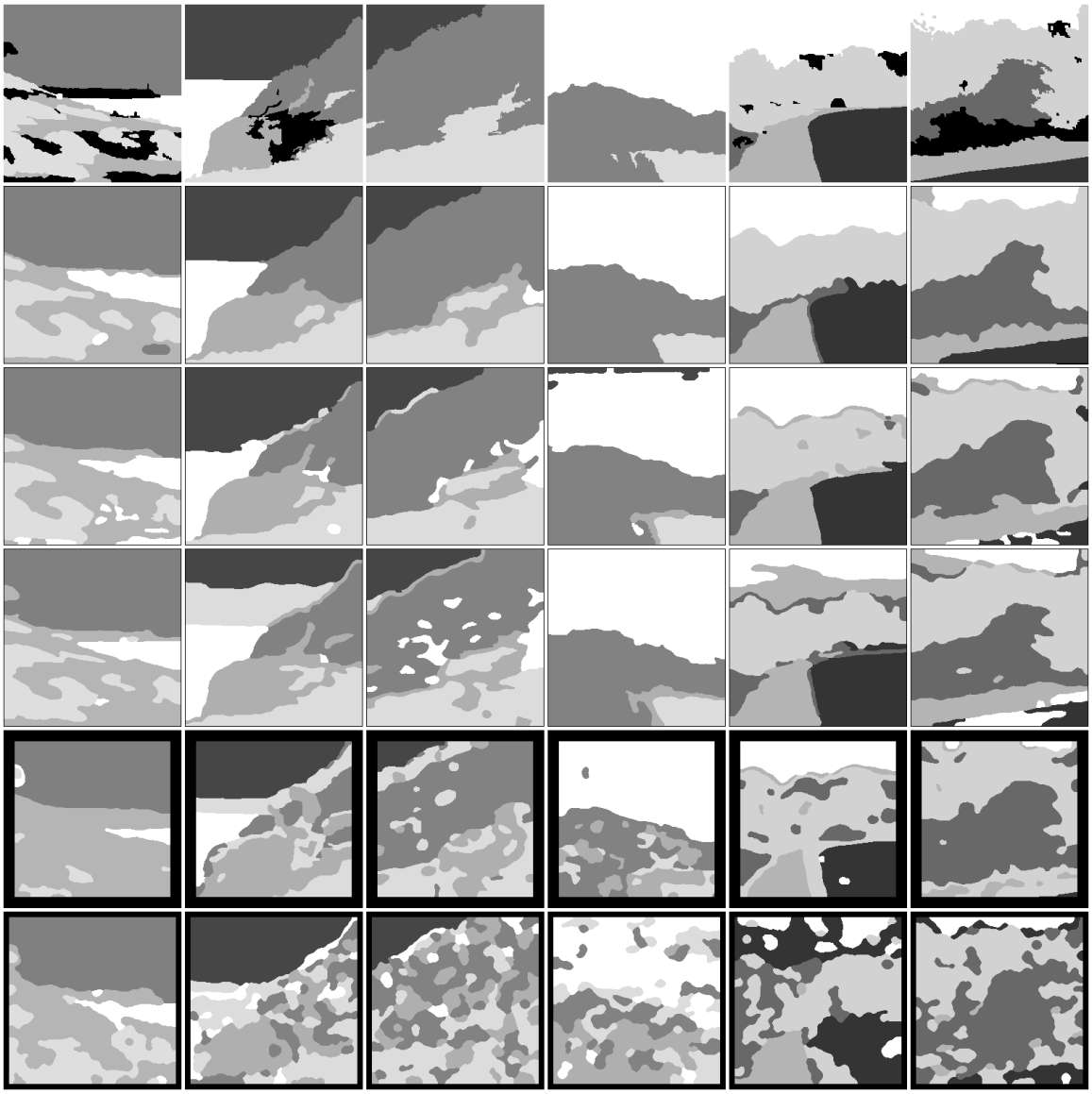

Figure 5.3: Classification maps for images 15, 19, 20, 21, 22 and 23. First row: ground-truth. Starting from the second row, results by: GMM-RDC-KNN+PR, GMM-IMMW, GMM-RDC-KNN′, MeasTex and LBP-G.

the approaches that utilize Gabor-based features obtain unsatisfactory results. Apart from these two images (and also image 5 to a lower extent), the proposed classifier is clearly superior to LBP-G, especially for the real scenes, where the latter yields very low classification rates and very little coherent segmentations, as shown in most of

UNIVERSITAT ROVIRA I VIRGILI
SUPERVISED AND UNSUPERVISED SEGMENTATION OF TEXTURED IMAGES BY EFFICIENT MULTI-LEVEL PATTERN CLASSIFICATION
Jaime Christian Meléndez Rodríguez
ISBN:978-84-693-7671-3/DL:T-1750-2010

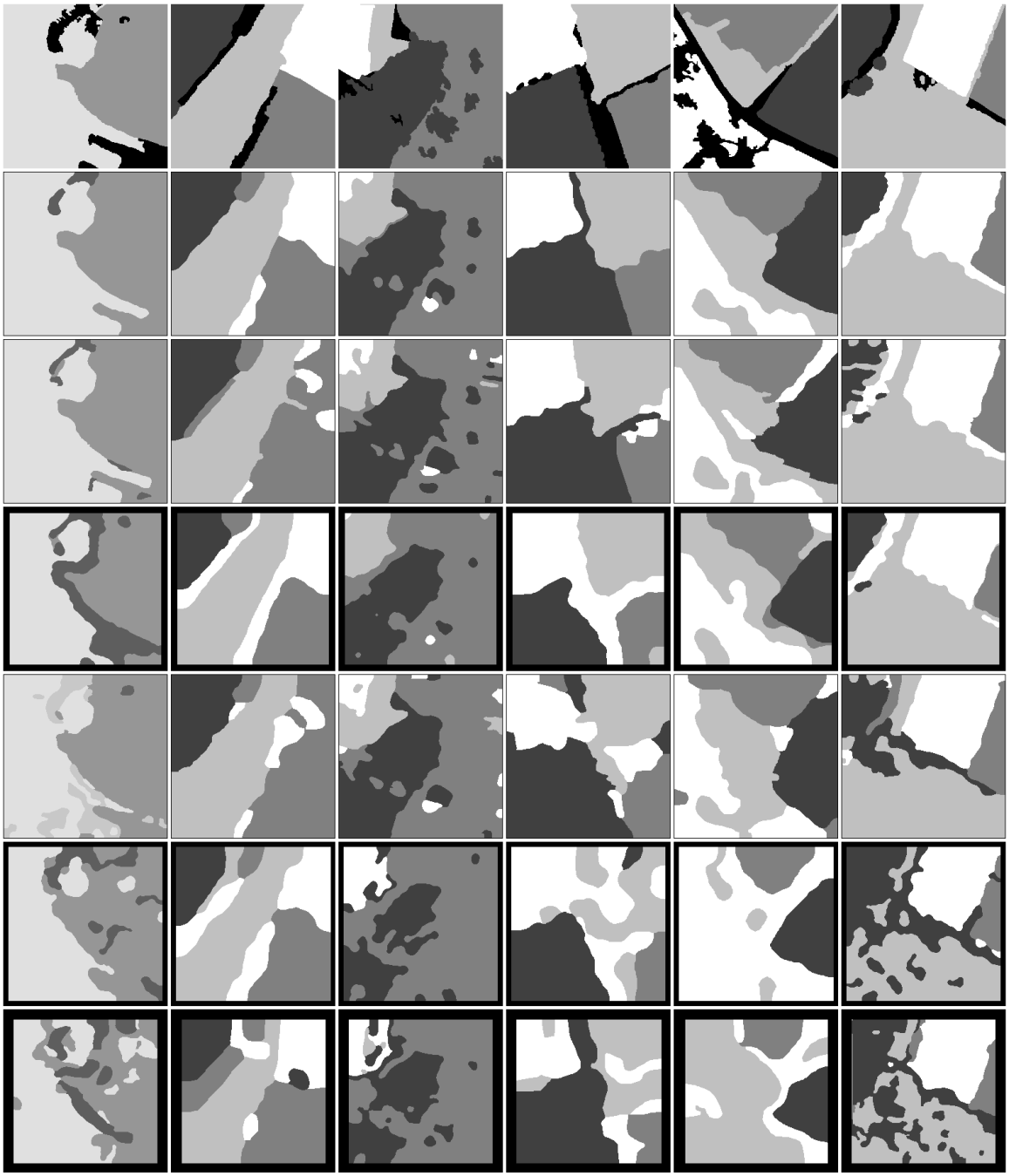

Figure 5.4: Classification maps for images 24, 26, 27, 28, 29 and 30. First row: ground-truth. Starting from the second row, results by: GMM-RDC-KNN+PR, GMM-RDC-KNN′, SVM-NN, GMM-IMMW, LBP-G and MeasTex.

the images in Figures 5.3 and 5.4.

Compared to the other multi-window approaches, GMM-RDC-KNN′ and GMM-IMMW, the advantages of the top-down scheme followed by GMM-RDC-KNN+PR are clearly demonstrated by the difference in classification rates and by the quality of the classification maps. In fact, the best single-window version of GMM-RDC-KNN performs better than GMM-RDC-KNN′ and GMM-IMMW (see Table 3.4), which indicates that their multi-window strategies are not always appropriate. For instance, in image 19, the 33×33 window used by GMM-RDC-KNN+PR in the first classification level is able to correctly identify the "sky" pattern and then the remaining windows simply refine the boundaries. On the contrary, the fusion scheme followed by GMM-RDC-KNN′ only identifies half of the region, whereas the other half is erroneously assigned to the "ground" pattern. Note that even MeasTex is able to identify the "sky" pattern, since it uses a 32×32 window, and GMM-IMMW as well, since the largest windows receive the largest weights. Similar errors are also found in images 22, 23 and 26. On the other hand, the weighting scheme followed by GMM-IMMW tends to misclassify a considerable amount of pixels, not only at boundary zones, but also inside homogeneous regions. This occurs even for the synthetic compositions, for which large windows should work fine. Thus, this indicates that not only its multi-window strategy is not behaving well, but also its Bayesian classifier with conflict resolution.

In terms of processing time, all the proposed classifiers are, by far, the fastest (see also Table 3.4). Regarding GMM-RDC-KNN+PR, its processing time is more than 30 times lower than the one achieved by LBP-G, which is the second best approach in terms of classification rates for the synthetic compositions, and more than 15 times lower than the one achieved by GMM-RDC-KNN′, which is the second best approach for the outdoor scenes and also the fastest alternative classifier.

The reasons for the low computation time achieved by GMM-RDC-KNN+PR are the following:

1. The reduced number of prototypes, which is related to the heuristic for efficiency maximization utilized by the proposed parameter selection algorithm. A comparison among the techniques that evaluate representative elements of the texture patterns, such as prototypes and SVs, reveals that, while the proposed technique produced an average of 208.47 prototypes per test image, Gabor-RDC-KNN′ produced around 2755.30 prototypes and SVM-NN produced around 5295.43 SVs. Furthermore, regarding the latter, it is interesting to note that the number of SVs substantially increases according to the difficulty of the problem. For instance, for image 3, which may be considered as an easy case according to the classification rates obtained for it, SVM-NN produced 2183 SVs. However, for image 4, which may be considered as a more difficult case, it produced 7733 SVs. Thus, a substantial increment in the computation time is expected as the separability among texture patterns decreases. On the contrary, the proposed methodology produced 207 and 305 prototypes, respectively, thus being more resilient to this problem.

2. The efficient utilization of multiple evaluation window sizes during classification by following a top-down approach, since only those pixels belonging to boundary zones are classified each time a different window size is applied. While GMM-RDC-KNN′ and GMM-IMMW perform 247124 pixel classifications considering a test image of 256×256 pixels and the four smallest window sizes, the proposed technique performs an average of 110780 pixel classifications for the same image size and an average of 3.77 windows, which means that more than half of the computation time is saved.

3. The use of low cost and simple classification techniques, such as the KNN rule (which in almost all situations reduces to the simplest nearest neighbor classifier ($K = 1$)) with the Euclidean distance as a distance measure. For instance, although both GMM-RDC-KNN+PR and LBP-G use similar classification tech-

niques and evaluate a similar number of prototypes or models, the computation of the G-statistic is much more expensive than the computation of the Euclidean distance, which leads to a significant difference in processing time.

5.3 Experiments on Unsupervised Texture Segmentation

The second set of experiments aims at validating the proposed methodology for unsupervised texture segmentation by performing a comparison with other unsupervised texture segmenters according to the experimental setup described in Section 4.3.1. Experiments are carried out in two parts, which are analogous to the ones reported in Sections 4.3.4 and 4.3.5. The proposed technique is referred to as GMM-PD$_{\text{GC}}$-PBC. Details about the alternative evaluated techniques are given below:

1. The segmenter proposed in (Rao et al., 2009) is based on the assumption that the optimal segmentation is the one that minimizes the overall coding length of the encoded textures and boundaries subject to a given distortion. This distortion parameter controls the level of detail of the segmentation. Texture features are constructed by taking pixels of each of the three $L^*a^*b^*$ channels over a hierarchy of windows of multiple sizes which, after convolution with a Gaussian kernel, are stacked into feature vectors and projected along their eight first principal components. Four window sizes are taken into account: 7×7, 5×5, 3×3 and 1×1. A pre-processing step consists of segmenting the image into several homogeneous regions known as superpixels. Later, after clustering, a post-processing step merges small regions. This segmentation technique is referred to as TBES.

2. The approach described in (Arbeláez et al., 2009) first applies an oriented watershed transform to form initial regions from a contour detector output and

then constructs an ultrametric contour map that defines a hierarchical segmentation, from which, for a given threshold, the output is a set of closed contours. The gPb detector (Maire et al., 2008), which combines multi-scale brightness, color and texture gradients, is applied as the initial step. This segmenter is referred to as OWT-UCM.

3. The compression-based texture merging algorithm described in (Yang et al., 2008) is based on the same methodology as TBES, but lacks boundary encoding and only utilizes a 7×7 window for texture feature extraction. This segmentation technique is referred to as CTM.

4. The segmentation technique proposed in (Muneeswaran et al., 2006) utilizes a combination of spatial and spectral distributions evaluated over a square neighborhood as the texture features that characterize each image pixel. The spectral distribution (spectral histograms) is derived from the Gabor filter bank discussed in Section 3.5.2. The number of bins per channel is set to 8, which means that the processed feature vectors are 200-dimensional. The size of the evaluation window is set to 17×17. These are the values that yield the best results over the entire image database. A post-processing procedure similar to the first step of the one described in Section 3.5.2 is defined by the authors, although it uses a 15×15 smoothing window. This segmentation technique is referred to as GMM-SSS.

5. The adaptive mean shift algorithm proposed in (Georgescu et al., 2003) exploits locally-sensitive hashing in order to reduce the computational complexity required to process high-dimensional features. In this implementation, the optimal parameters of the data structure are determined by a pilot learning procedure. Therefore, the only free parameter is the number of neighbors evaluated by this procedure. This parameter controls the resolution of the data analysis. Texture features for this technique are derived from the Gabor filter bank

discussed in Section 3.5.2 and correspond to the 17×17 evaluation window, which is the one that yields the best results for the whole test set. Since this algorithm is not a complete segmentation technique, the post-processing stage described in Section 3.5.2 is included. The resulting segmenter is referred to as GMM-LSHAMS.

6. EdgeFlow (Ma and Manjunath, 2000) is a texture segmentation scheme based on boundary detection that utilizes a predictive coding model to identify the direction of change in intensity and texture at each image location for a given scale. The scale parameter is defined by the user and controls the level of detail of the segmentation. Texture features are obtained by means of a Gabor filter bank that follows the design strategy proposed in (Manjunath and Ma, 1996). A post-processing step that aims at connecting disjoint boundaries and merging small regions is defined by the authors. This segmentation technique is referred to as EdgeFlow.

First Part

In the first part, the best outputs for each test case in terms of segmentation quality are compared. Therefore, the parameter that controls the level of segmentation detail of each technique is set to be optimal for each test case as in Section 4.3.4. In addition to the segmentation quality, the processing time is also assessed. Processing times correspond to an Intel Pentium IV at 3.2 GHz with 2 GB of RAM, except for CTM, OWT-UCM and TBES, which correspond to an Intel Core 2 Quad at 2.4 GHz with 4 GB of RAM. It is important to mention that, since their public implementations are written in Matlab and C (mex files), the computation time corresponding to the Matlab part, which is roughly 55, 70 and 80 percent of the total for each technique, could be improved if they were entirely written in C. Finally, since those algorithms are also designed to work with color features, results for both gray-scale and color images (Figures A.3, A.5, A.7 and A.9) are given.

UNIVERSITAT ROVIRA I VIRGILI
SUPERVISED AND UNSUPERVISED SEGMENTATION OF TEXTURED IMAGES BY EFFICIENT MULTI-LEVEL PATTERN CLASSIFICATION
Jaime Christian Meléndez Rodríguez
ISBN:978-84-693-7671-3/DL:T-1750-2010

Table 5.2: Average segmentation quality ($\overline{Q}$) and average processing time ($\overline{PT}$) for the approaches evaluated in the first part of the second set of experiments.

Segmentation	$\overline{Q}$						$\overline{PT}$
technique	set a	set b	set c	set d	set e	all sets	all sets
GMM-PD$_{GC}$-PBC	96.29	89.43	86.59	90.43	89.74	89.72	4.272
TBES	39.63	26.46	85.20	83.64	81.25	70.07	494.854
TBES (color)	—	85.37	83.54	84.41	79.92	82.72	489.613
OWT-UCM	58.50	43.20	82.54	83.96	84.32	74.89	131.842
OWT-UCM (color)	—	54.83	83.24	80.28	86.66	80.21	132.341
CTM	26.45	23.03	76.80	66.89	80.97	62.73	135.293
CTM (color)	—	54.28	81.07	79.89	89.73	80.37	127.242
GMM-SSS	84.76	80.58	78.29	80.71	79.34	80.11	4.861
GMM-LSHAMS	79.35	72.04	81.45	83.11	76.71	78.84	598.032
EdgeFlow	73.67	70.17	78.28	81.13	73.31	75.62	16.411

Average results for this comparison are summarized in Table 5.2. Figures 5.5 to 5.9 show some segmentation maps produced by the top performing approaches for each image set. These results indicate that the proposed segmentation technique obtains higher quality scores than the other approaches. In fact, taking into account all sets, the other versions of the proposed technique, GMM-PD$_{KMC}$-PBC and GMM-PD$_{MSC}$-PBC, also obtain higher scores (see Table 4.1). In particular, the version evaluated here, GMM-PD$_{GC}$-PBC, is superior to the closest alternative approach, TBES (color), by seven points, even when the texture features for the latter are obtained from three color channels instead of the single gray-level channel processed by GMM-PD$_{GC}$-PBC. Therefore, the proposed technique produces, in general, better segmentations considering less information. Moreover, when TBES is run on gray-scale images, its performance degrades by around sixty points for the synthetic compositions, but, surprisingly, slightly increases for the outdoor scenes, which clearly indicates that the success of this technique highly depends on the intensity differences of the considered

UNIVERSITAT ROVIRA I VIRGILI
SUPERVISED AND UNSUPERVISED SEGMENTATION OF TEXTURED IMAGES BY EFFICIENT MULTI-LEVEL PATTERN CLASSIFICATION
Jaime Christian Meléndez Rodríguez
ISBN:978-84-693-7671-3/DL:T-1750-2010

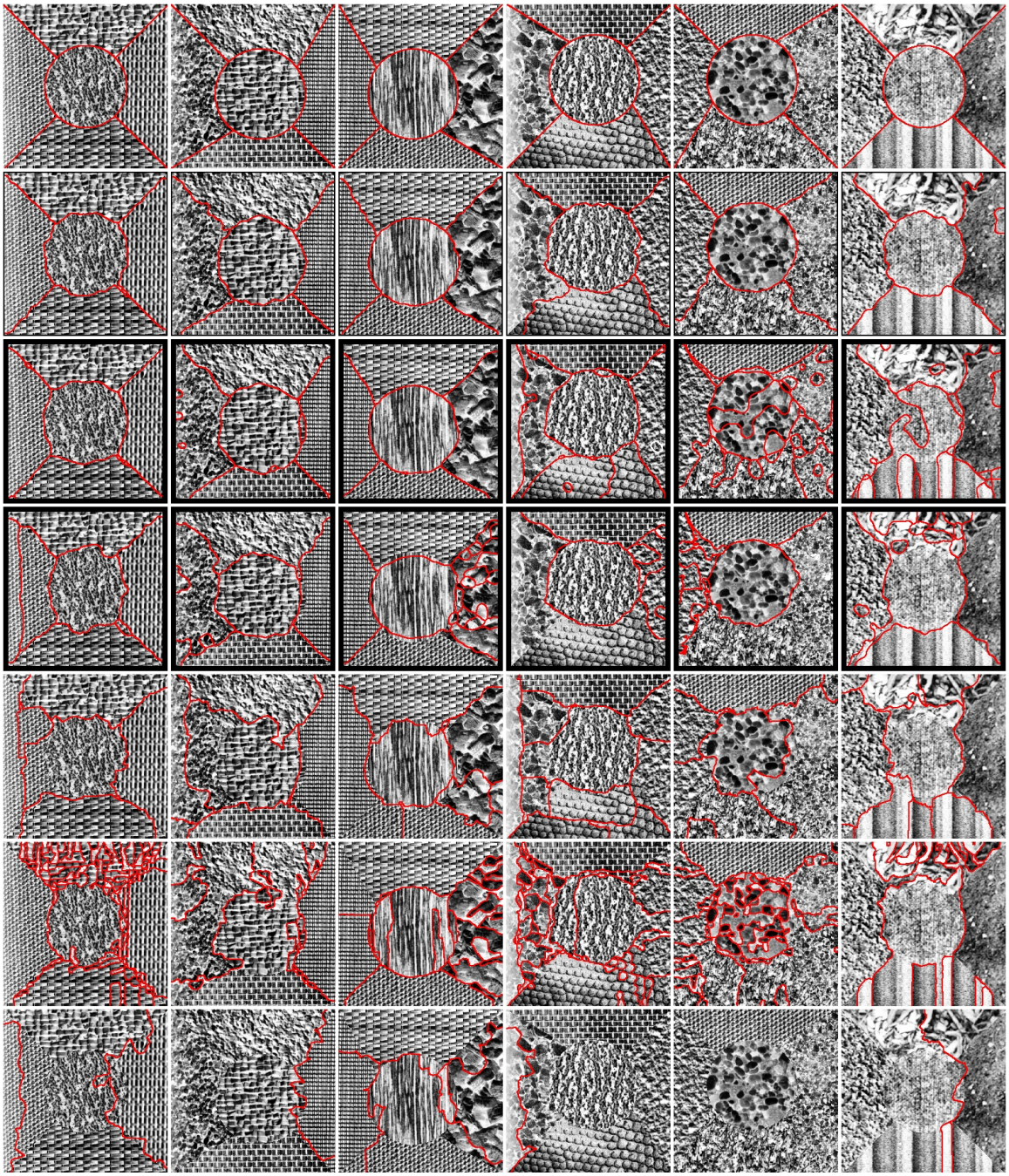

Figure 5.5: Segmentation maps for images 1, 2, 3, 4, 5 and 7. First row: ground-truth. Starting from the second row, results by: GMM-PD$_{GC}$-PBC, GMM-SSS, GMM-LSHAMS, EdgeFlow, OWT-UCM and TBES.

UNIVERSITAT ROVIRA I VIRGILI
SUPERVISED AND UNSUPERVISED SEGMENTATION OF TEXTURED IMAGES BY EFFICIENT MULTI-LEVEL PATTERN CLASSIFICATION
Jaime Christian Meléndez Rodríguez
ISBN:978-84-693-7671-3/DL:T-1750-2010

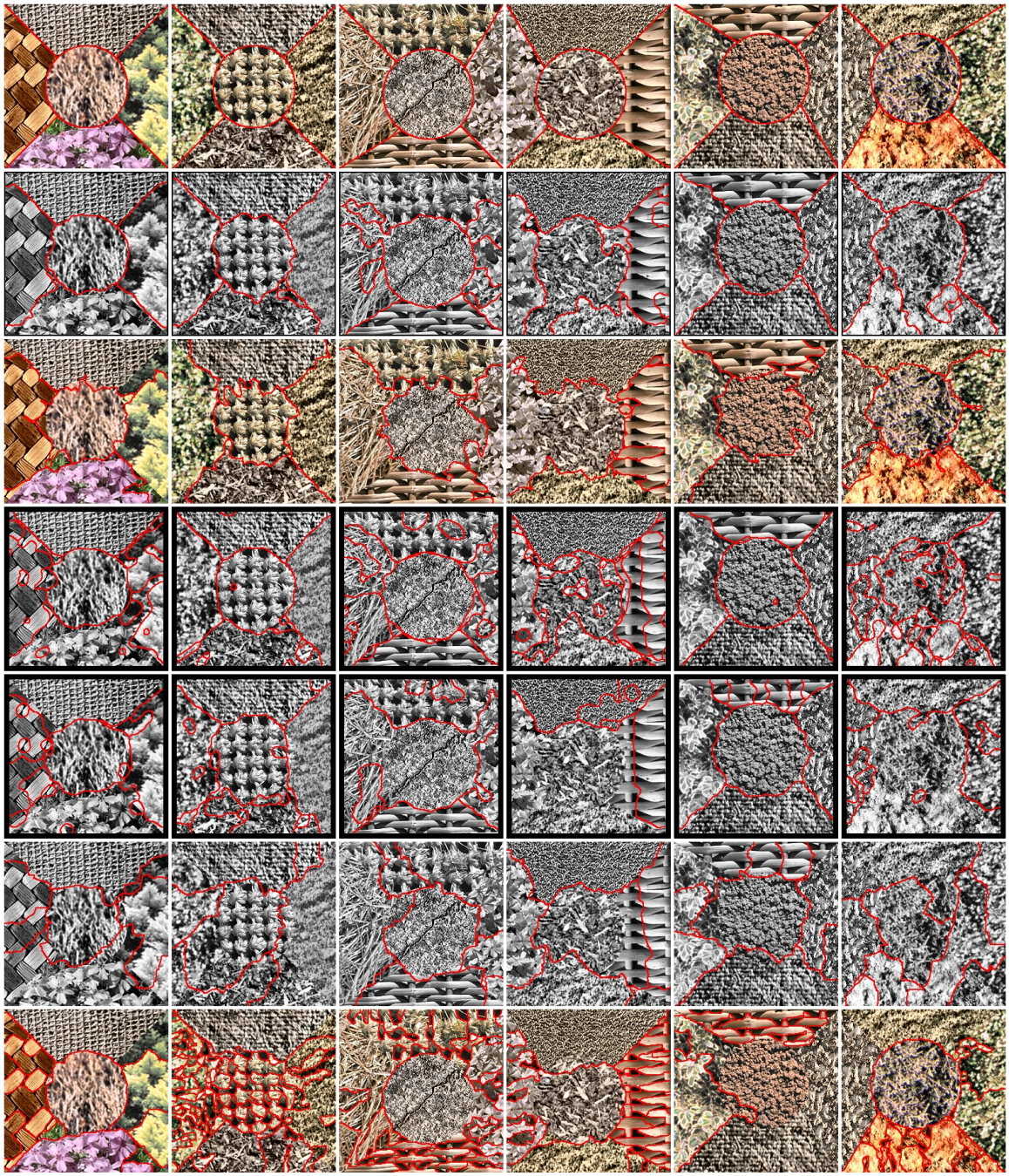

Figure 5.6: Segmentation maps for images 8, 10, 11, 12, 13 and 14. First row: ground-truth. Starting from the second row, results by: GMM-PD_{GC}-PBC, TBES (color), GMM-SSS, GMM-LSHAMS, EdgeFlow and OWT-UCM (color).

UNIVERSITAT ROVIRA I VIRGILI
SUPERVISED AND UNSUPERVISED SEGMENTATION OF TEXTURED IMAGES BY EFFICIENT MULTI-LEVEL PATTERN CLASSIFICATION
Jaime Christian Meléndez Rodríguez
ISBN:978-84-693-7671-3/DL:T-1750-2010

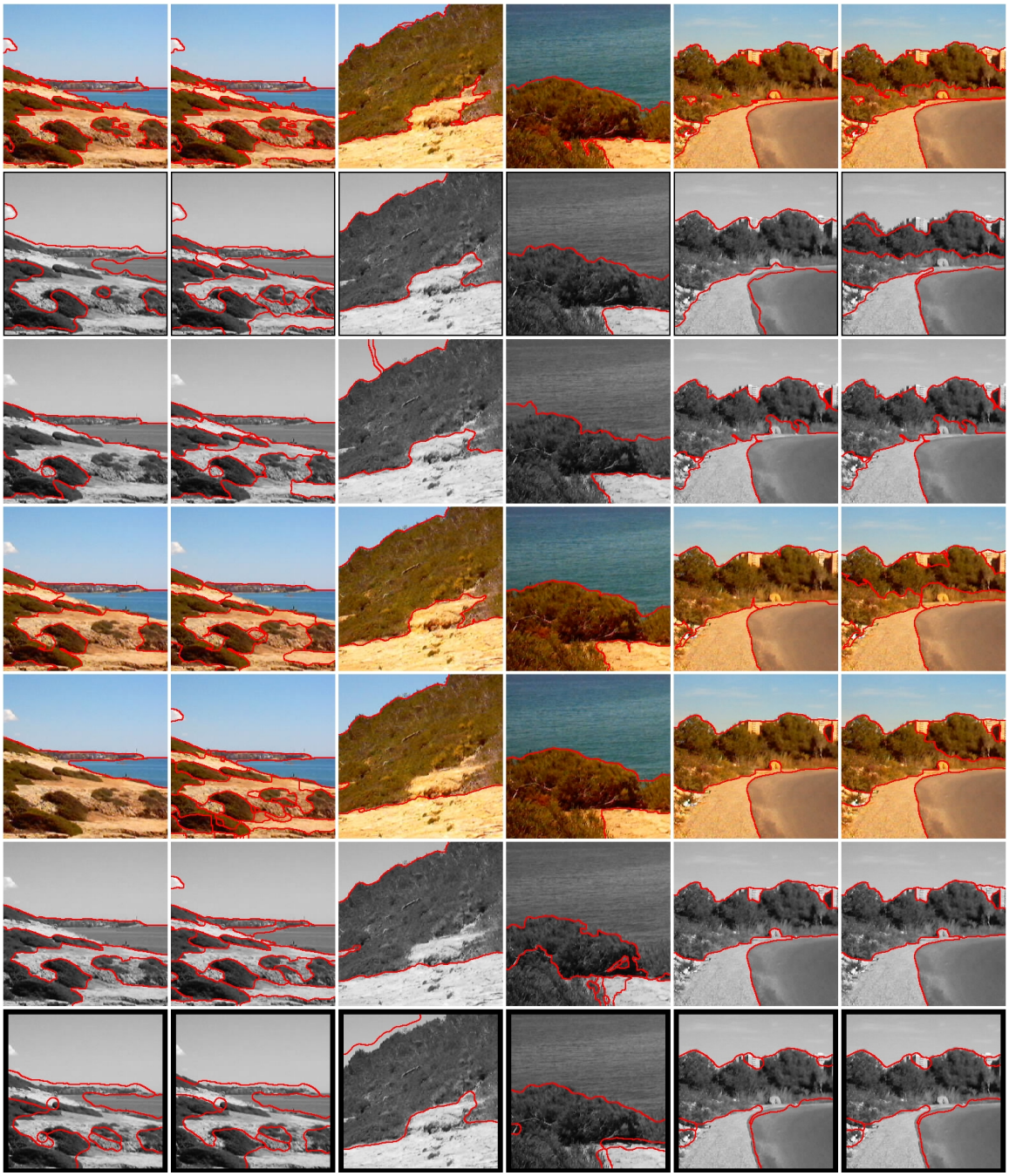

Figure 5.7: Segmentation maps for images 15, 20, 21 and 22. First row: ground-truth. Starting from the second row, results by: GMM-PD$_{GC}$-PBC, TBES, TBES (color), OWT-UCM (color), OWT-UCM and GMM-LSHAMS.

UNIVERSITAT ROVIRA I VIRGILI
SUPERVISED AND UNSUPERVISED SEGMENTATION OF TEXTURED IMAGES BY EFFICIENT MULTI-LEVEL PATTERN CLASSIFICATION
Jaime Christian Meléndez Rodríguez
ISBN:978-84-693-7671-3/DL:T-1750-2010

Figure 5.8: Segmentation maps for images 26 to 31. First row: ground-truth. Starting from the second row, results by: GMM-PD$_{GC}$-PBC, TBES (color), OWT-UCM, TBES, GMM-LSHAMS and EdgeFlow.

UNIVERSITAT ROVIRA I VIRGILI
SUPERVISED AND UNSUPERVISED SEGMENTATION OF TEXTURED IMAGES BY EFFICIENT MULTI-LEVEL PATTERN CLASSIFICATION
Jaime Christian Meléndez Rodríguez
ISBN:978-84-693-7671-3/DL:T-1750-2010

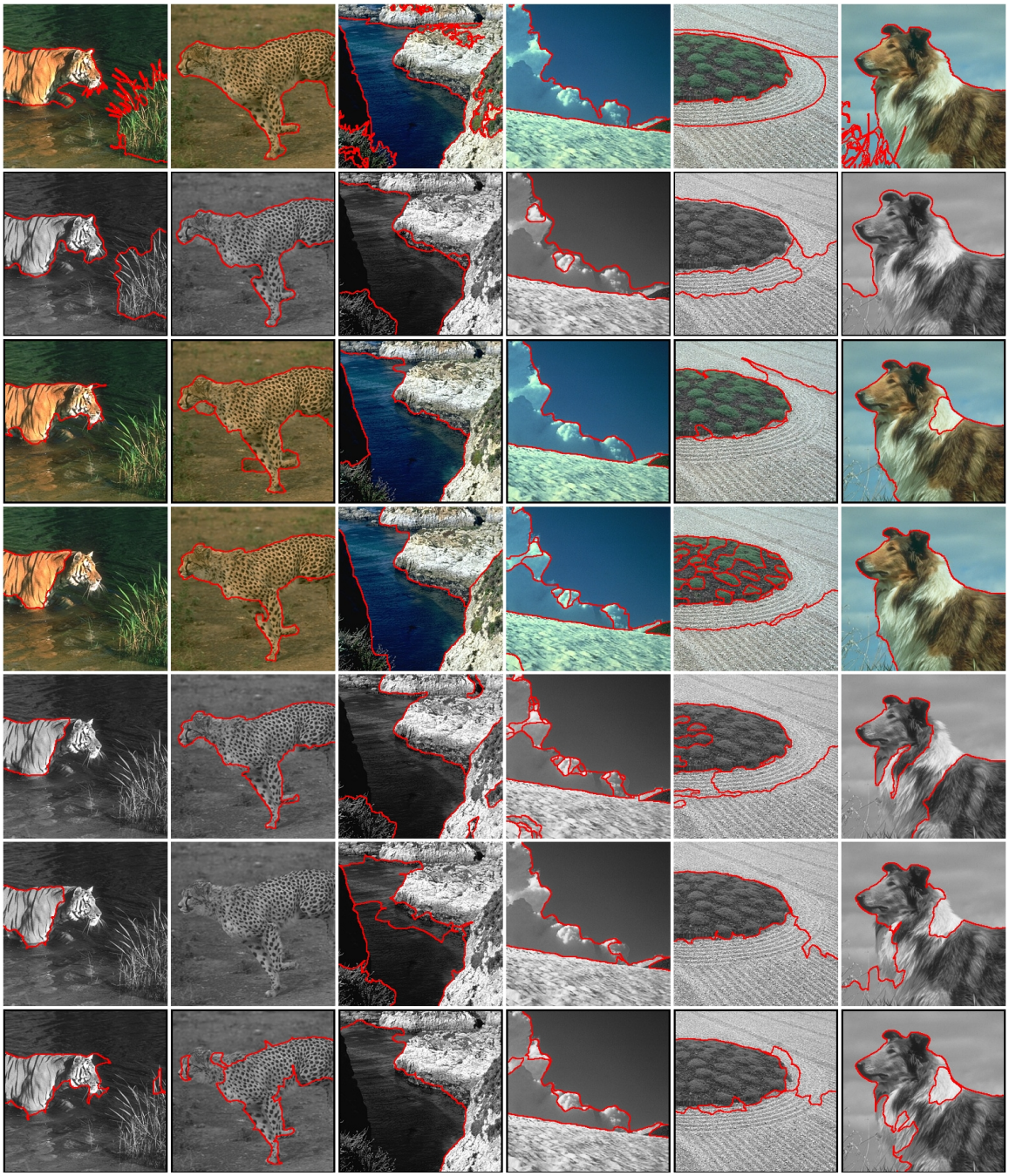

Figure 5.9: Segmentation maps for images 34, 36, 37, 39, 45 and 46. First row: ground-truth. Starting from the second row, results by: GMM-PD$_{GC}$-PBC, CTM (color), OWT-UCM (color), OWT-UCM, TBES and CTM.

texture patterns, as those differences are more noticeable in the gray-scale versions of the outdoor scenes. This observation also applies to OWT-UCM and, in a different way, to CTM, since it experiences degradation in all types of images when run on gray-scale images.

In terms of processing time, all versions of the proposed technique are the fastest (see Table 4.1 again). For instance, compared with TBES (color), GMM-PD$_{\mathrm{GC}}$-PBC is more than one hundred times faster. Therefore, the superior segmentation quality discussed above comes along with a considerable saving of computation time. The reasons for that low execution time are:

1. The efficient application of multiple evaluation windows following a top-down approach during the pixel-based classification stage, as discussed in Section 5.2.

2. The use of a fast SVM-based classifier, both in training and classification, which introduces a minimum overhead in the whole segmentation process.

3. The reduced number of feature vectors processed by the pattern discovery stage, as only a rough approximation of the image's texture patterns is needed for obtaining a good final segmentation.

Second Part

In the second part, a robustness test, such as the one reported in Section 4.3.5, is carried out. Therefore, the parameter that controls the level of segmentation detail of each evaluated technique is kept fixed along the entire image sequence, whereas the remaining parameters are set as in the previous part. Only the color versions of TBES, OWT-UCM and CTM are considered. The performance of the tested algorithms is measured in terms of segmentation quality.

Table 5.3 shows the segmentation quality scores resulting from this test. Figures 5.10 and 5.11 show the segmentation maps produced by the top performing approaches. Again, all versions of the proposed technique obtain the highest average

UNIVERSITAT ROVIRA I VIRGILI
SUPERVISED AND UNSUPERVISED SEGMENTATION OF TEXTURED IMAGES BY EFFICIENT MULTI-LEVEL PATTERN CLASSIFICATION
Jaime Christian Meléndez Rodríguez
ISBN:978-84-693-7671-3/DL:T-1750-2010

Table 5.3: Segmentation quality (Q) for the approaches evaluated in the second part of the second set of experiments.

Segmentation technique	type of segmentation	Q per image				$\overline{Q}$
		15	16	17	18	
GMM-PD$_{GC}$-PBC	coarse	86.51	86.89	87.85	83.51	86.19
	fine	86.28	82.55	85.33	74.89	82.26
TBES (color)	coarse	71.58	72.51	82.69	79.02	76.45
	fine	77.07	70.05	69.74	73.33	72.55
OWT-UCM (color)	coarse	77.02	68.04	90.75	65.86	75.42
	fine	68.28	64.18	82.43	75.87	72.69
CTM (color)	coarse	74.77	88.44	90.88	68.01	80.53
	fine	68.75	73.30	80.71	43.82	66.65
GMM-SSS	coarse	81.81	81.50	83.81	82.07	82.30
	fine	64.67	60.49	72.81	58.74	64.18
GMM-LSHAMS	coarse	85.99	76.74	84.38	54.11	75.31
	fine	74.55	70.15	77.35	74.77	74.21
EdgeFlow	coarse	68.60	63.82	60.93	60.97	63.58
	fine	83.81	64.21	62.07	75.33	71.36

scores for both coarse and fine segmentations (see also Table 4.3). It is interesting to note that the same difficulty regarding the fine segmentation of image 18 discussed for the proposed techniques in Section 4.3.5 is also experienced here by the alternative segmentation techniques.

Among the evaluated approaches, GMM-PD$_{GC}$-PBC is again the most robust, since it is the one that best combines stability and high segmentation quality. When compared with the alternative approaches, it is only surpassed by OWT-UCM (color), CTM (color) and Edgeflow in few individual cases. However, the latter achieve considerably lower average scores, especially in the fine segmentation problem.

UNIVERSITAT ROVIRA I VIRGILI
SUPERVISED AND UNSUPERVISED SEGMENTATION OF TEXTURED IMAGES BY EFFICIENT MULTI-LEVEL PATTERN CLASSIFICATION
Jaime Christian Meléndez Rodríguez
ISBN:978-84-693-7671-3/DL:T-1750-2010

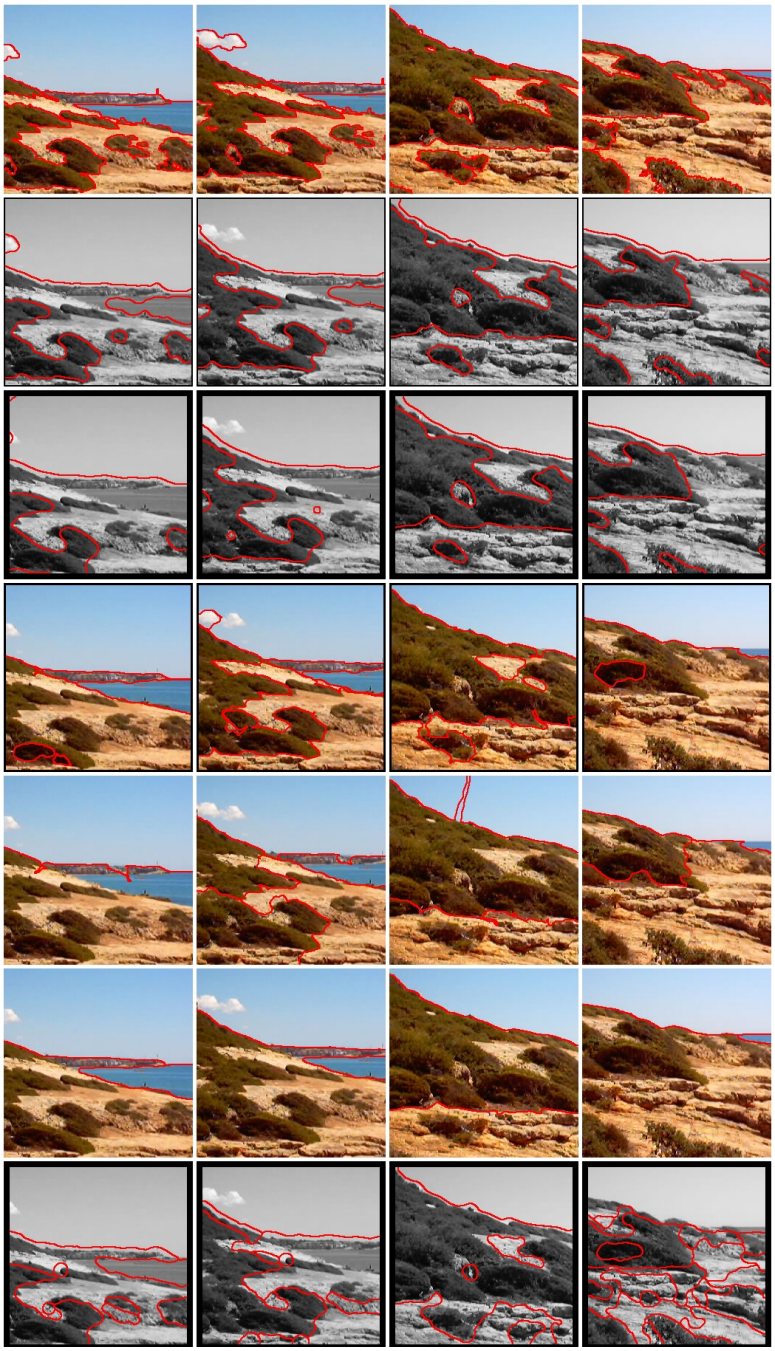

Figure 5.10: Segmentation maps for images 15 to 18 (coarse segmentation). First row: ground-truth. Starting from the second row, results by: GMM-PD_{GC}-PBC, GMM-SSS, CTM (color), TBES (color), OWT-UCM (color) and GMM-LSHAMS.

UNIVERSITAT ROVIRA I VIRGILI
SUPERVISED AND UNSUPERVISED SEGMENTATION OF TEXTURED IMAGES BY EFFICIENT MULTI-LEVEL PATTERN CLASSIFICATION
Jaime Christian Meléndez Rodríguez
ISBN:978-84-693-7671-3/DL:T-1750-2010

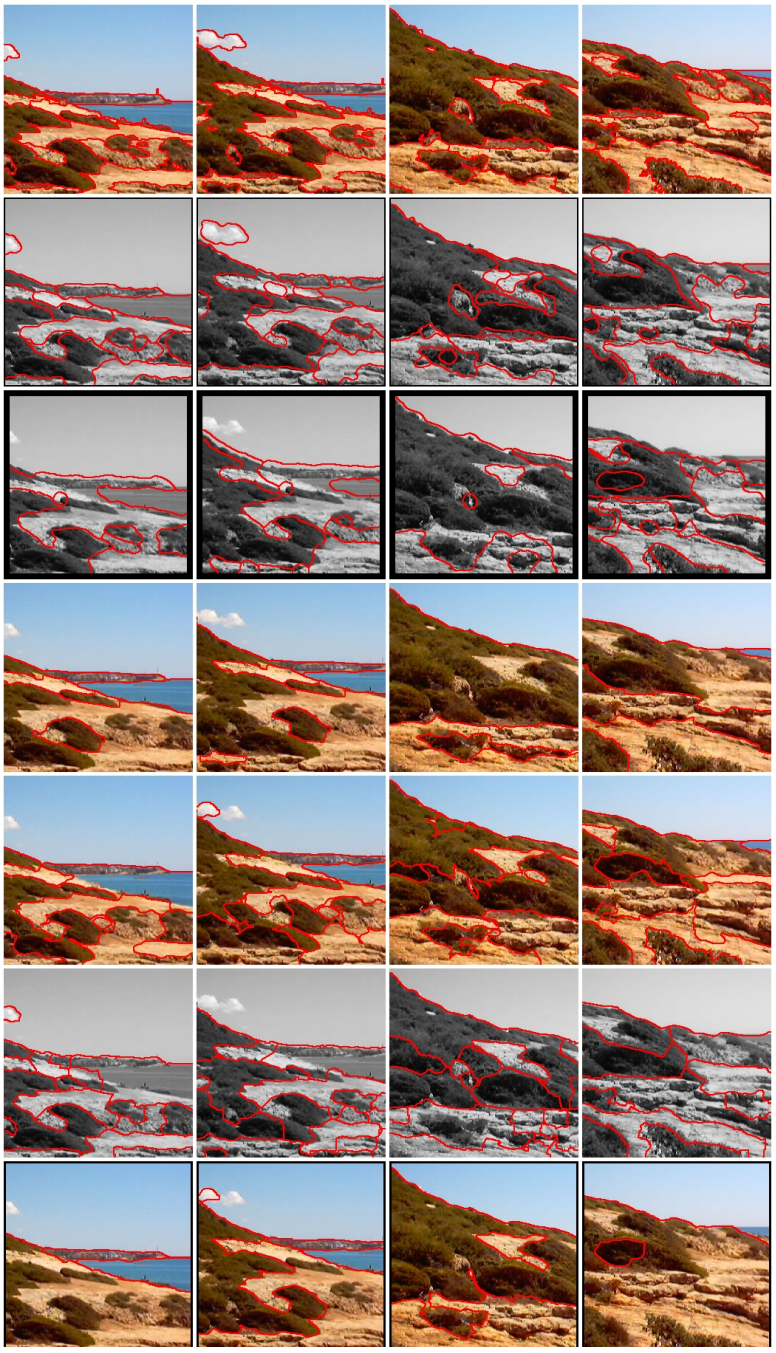

Figure 5.11: Segmentation maps for images 15 to 18 (fine segmentation). First row: ground-truth. Starting from the second row, results by: GMM-PD$_{GC}$-PBC, GMM-LSHAMS, OWT-UCM (color), TBES (color), EdgeFlow and CTM (color).

5.4 Comparisons Between Supervised and Unsupervised Techniques

It was stated in Section 4.1 that supervised segmentation algorithms usually obtain more accurate results than their unsupervised counterparts, and this is precisely the reason that motivated the two-stage methodology for unsupervised texture segmentation proposed in this thesis. Thus, in the last set of experiments, the techniques developed for both texture application domains, namely: GMM-SVM-OAO+SVR and GMM-PD$_{\mathrm{GC}}$-PBC, are compared. Although it is not possible to compute classification rates for GMM-PD$_{\mathrm{GC}}$-PBC, as explained in Section 4.3.1, it is possible to compute Q values for GMM-SVM-OAO+SVR. Therefore, the segmentation quality factor defined for the unsupervised case is utilized as an accuracy measure.

Only image sets a and b have been considered in this experiment, since they are the unique sets for which, in all of their instances, all texture classes present in the test image are also available in the training set supplied to the supervised classifier (recall the "unknown" classes, the absence of certain patterns and the different possible ground-truths for many of the real scenes). The compared techniques are configured according to Sections 3.5.2 (including the SV reduction method) and 4.3.2.

The results for this experiment are given in Table 5.4. Two important conclusions can be derived from them:

1. By comparing the average Q values for GMM-SVM-OAO+SVR with those for the alternative unsupervised segmentation techniques of the previous section (see Table 5.2), the earlier statement about the advantage of the supervised segmentation approaches over the unsupervised ones is corroborated, as all the alternative techniques yield lower scores.

2. GMM-PD$_{\mathrm{GC}}$-PBC has appropriately incorporated the desired supervision capabilities, as it achieves a higher average score than GMM-SVM-OAO+SVR. In

UNIVERSITAT ROVIRA I VIRGILI
SUPERVISED AND UNSUPERVISED SEGMENTATION OF TEXTURED IMAGES BY EFFICIENT MULTI-LEVEL PATTERN CLASSIFICATION
Jaime Christian Meléndez Rodríguez
ISBN:978-84-693-7671-3/DL:T-1750-2010

Table 5.4: Segmentation quality (Q) for the approaches evaluated in the third set of experiments.

	Q per technique	
	GMM-SVM-OAO+SVR	GMM-PD_{GC}-PBC
image 1	98.01	97.98
image 2	96.52	96.82
image 3	98.36	98.21
image 4	95.58	94.52
image 5	80.61	95.45
image 6	96.16	96.84
image 7	88.70	94.18
Avg. set a	93.42	96.29
image 8	92.25	96.13
image 9	94.51	94.43
image 10	95.80	95.68
image 11	88.02	78.81
image 12	83.15	83.91
image 13	97.17	96.09
image 14	75.74	80.97
Avg. set b	89.52	89.43
Avg. sets a-b	91.47	92.86

this line, the main advantage of GMM-PD_{GC}-PBC over GMM-SVM-OAO+SVR is that, whereas the latter uses a separate training set that does not exactly match with the patterns in the test image, the former takes its texture models directly from the test image. The segmentation maps for both of the evaluated techniques are shown in Figure 5.12.

UNIVERSITAT ROVIRA I VIRGILI
SUPERVISED AND UNSUPERVISED SEGMENTATION OF TEXTURED IMAGES BY EFFICIENT MULTI-LEVEL PATTERN CLASSIFICATION
Jaime Christian Meléndez Rodríguez
ISBN:978-84-693-7671-3/DL:T-1750-2010

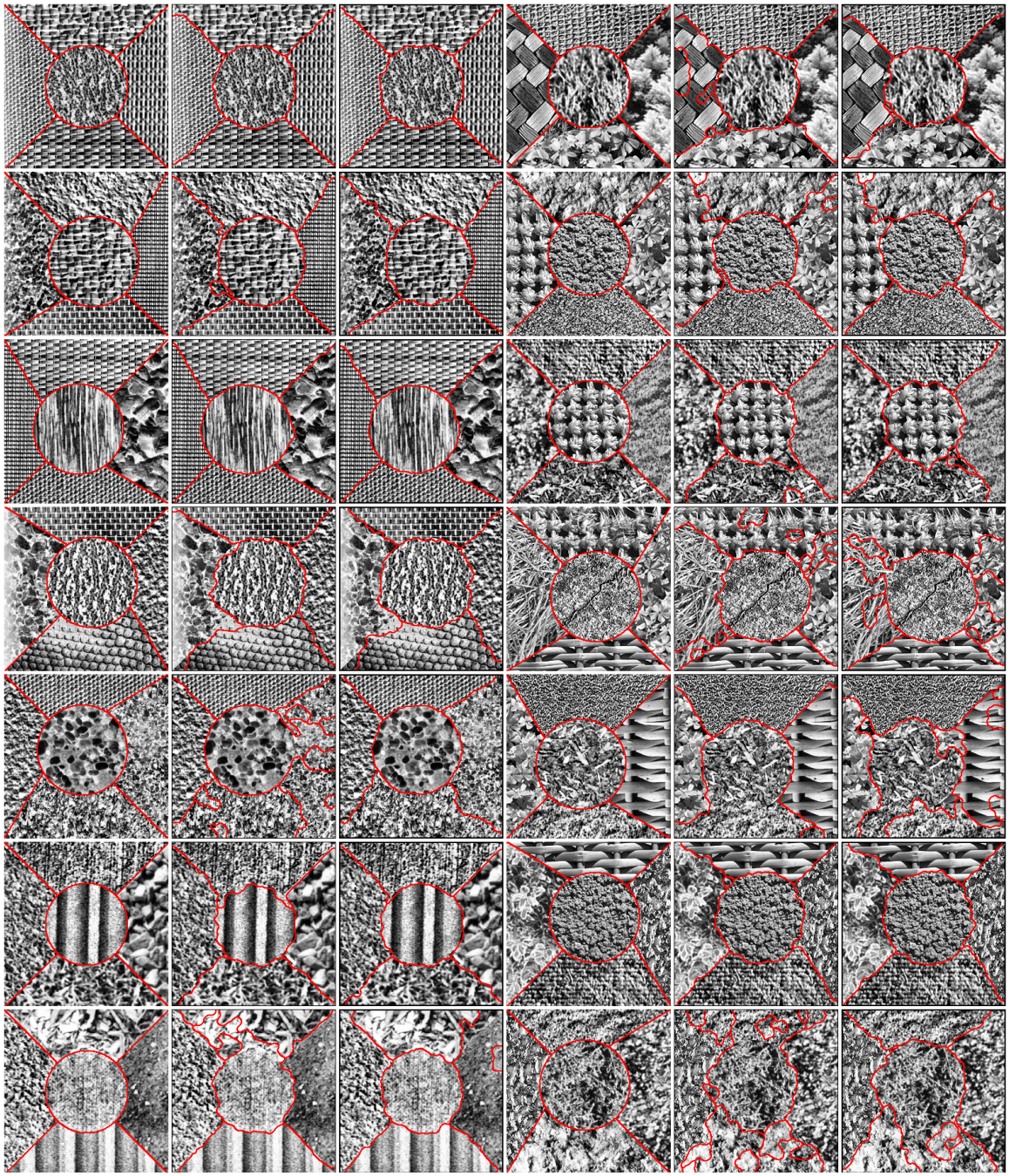

Figure 5.12: Segmentation maps for images 1 to 14. Corresponding ground-truth (first and fourth columns). Results by: GMM-SVM-OAO+SVR (second and fifth columns) and GMM-PD_{GC}-PBC (third and sixth columns).

5.5 Summary

This chapter presents the experimental validation of the supervised and unsupervised texture segmentation methodologies developed in Chapters 3 and 4, respectively. Three sets of experiments have been performed in order to complement those reported in the previous chapters.

In the first set of experiments, the best classifier in terms of classification rates developed in Chapter 3 has been compared with different well-known supervised texture classifiers. The obtained results show that the proposed technique yields higher classification rates than the other approaches with significantly lower processing times, due to its efficient classification scheme.

In the second set of experiments, the best segmentation technique in terms of segmentation quality developed in Chapter 4 has been compared with alternative well-known unsupervised texture segmenters. The results in the first part of these experiments show that the proposed technique produces the best segmentation quality and is the fastest among the evaluated approaches. The results in the second part show that the proposed technique is the most robust, as it achieves higher average segmentation quality when it is evaluated along a sequence of images while keeping its parameter configuration invariable.

Finally, in the last set of experiments, both the supervised and unsupervised methodologies developed in this thesis have been compared in terms of segmentation quality. As a result, it has been shown that the proposed supervised segmentation technique outperforms a number of unsupervised segmenters previously reported in the literature. In addition, it has also been shown that the extension of the proposed technique to the unsupervised domain is, indeed, advantageous, as the segmentation quality yielded by this extension is, on average, superior to the one yielded by the supervised version.

UNIVERSITAT ROVIRA I VIRGILI
SUPERVISED AND UNSUPERVISED SEGMENTATION OF TEXTURED IMAGES BY EFFICIENT MULTI-LEVEL PATTERN CLASSIFICATION
Jaime Christian Meléndez Rodríguez
ISBN:978-84-693-7671-3/DL:T-1750-2010

Chapter 6

Concluding Remarks

This chapter presents the final remarks of this dissertation. It is organized as follows. Section 6.1 summarizes the contents of this dissertation and its main contributions. Section 6.2 proposes future lines of research that arise from the development of this thesis. Finally, Section 6.3 lists the publications that have already been derived from this thesis.

6.1 Summary and Contributions

Improving low-level tasks, such as image segmentation and classification based on texture, has been the main objective of this thesis, which has led to several contributions in the fields of computer vision and pattern recognition as highlighted below.

6.1.1 Supervised Pixel-Based Texture Classification for Supervised Texture Segmentation

A new methodology for evaluating and processing texture information has been proposed with the final purpose of improving supervised pixel-based texture classification as a way to perform supervised texture segmentation. The classification scheme pre-

UNIVERSITAT ROVIRA I VIRGILI
SUPERVISED AND UNSUPERVISED SEGMENTATION OF TEXTURED IMAGES BY EFFICIENT MULTI-LEVEL PATTERN CLASSIFICATION
Jaime Christian Meléndez Rodríguez
ISBN:978-84-693-7671-3/DL:T-1750-2010

sented in this dissertation is based on two new stages that consist of processing texture features obtained from the evaluation of texture methods over multiple windows of different size and, on the other hand, of the efficient learning of texture models from these features.

6.1.1.1 Classification with Multiple Evaluation Windows

Usually, texture classifiers and segmenters evaluate texture methods over single windows whose size is set on an empirical basis. Since the performance of texture methods is greatly influenced by the particular textures to which they are applied, it is not possible to devise general strategies for determining single window sizes that allow optimal discrimination of arbitrary textures. Although there have been a few attempts to integrate the information of multiple evaluation window sizes, they still do not take full advantage of the multi-window paradigm.

Following a different approach from the current algorithms, this work proposes a methodology that consists of evaluating texture methods over multiple windows of different size and then classifying the image pixels according to a multi-level top-down scheme that starts with the largest available evaluation window and then refines the boundary zones of the current segmentation by sequentially applying the remaining windows. This methodology has proven to yield good results for segmentation and classification of images both within wide regions of uniform texture, where large windows are appropriate, as well as in boundary zones among different textures, where small windows are more adequate in order to identify the correct texture.

Experiments have demonstrated that, in general, the proposed scheme leads to higher classification rates than previously developed multi-window and single-window classifiers due to the benefits derived from the utilization of the aforementioned top-down approach.

6.1.1.2 Efficient Learning of Texture Models

Another important drawback of current image classifiers and segmenters based on texture features is that they do not provide an efficient characterization of the learnt texture patterns. Therefore, in order to achieve accurate results, computationally expensive strategies are utilized, which limit their application in tasks where the processing time is relevant.

In particular, due to the nature of pixel-based classification, a huge number of texture feature vectors are associated with the training patterns. Therefore, it is necessary to summarize all the available information in order to efficiently model each pattern. In this thesis, three efficient pixel-based classifiers are proposed. The first two classifiers utilize the KNN rule. Thus, their main issue is how to determine appropriate sets of prototypes that yield a high classification accuracy while keeping a low processing time. To solve this problem, prototype computation by a resolution-driven clustering algorithm and a normalized cut-based clustering algorithm has been introduced. For the third classifier, a one-against-one multiclass extension of binary SVMs that incorporates probability estimates in its voting strategy is utilized.

Experimental results presented in this dissertation have demonstrated that these efficient pixel-based classifiers lead to higher classification rates and significantly lower processing times than previous supervised classifiers proposed in the literature.

6.1.2 Automatic Parameter Selection

An automatic parameter selection algorithm is integrated in the classification methodology proposed in this thesis with the purpose of determining a suitable set of parameters for configuring the proposed classifiers. It consists of a search procedure that aims at identifying the combination of parameters that yield the best classification rates in classifying the training set. Three parameters have been considered: the number of neighbors used in the voting stage of the KNN classifier, the number of prototypes

UNIVERSITAT ROVIRA I VIRGILI
SUPERVISED AND UNSUPERVISED SEGMENTATION OF TEXTURED IMAGES BY EFFICIENT MULTI-LEVEL PATTERN CLASSIFICATION
Jaime Christian Meléndez Rodríguez
ISBN:978-84-693-7671-3/DL:T-1750-2010

(or a related parameter) evaluated during classification, and the maximum number of evaluation windows applied by the top-down scheme discussed earlier. While all these three parameters are taken into account to configure the KNN-based classifiers, only the last one is considered for the SVM-based classifier.

The conducted experiments indicate that this parameter selection algorithm is able to determine combinations of parameters that lead to "near-optimal" classification rates.

6.1.3 A Heuristic for Efficiency Improvement

A new version of a previously proposed heuristic for efficiency computation derived from the well-known benefit-cost ratio has been included in this thesis as a criterion for prototype reduction and feature selection.

6.1.3.1 Prototype Reduction

In this case, the automatic parameter selection algorithm summarized above is modified in order to search for the combination of parameters that yield the maximization of the newly proposed heuristic instead of the maximization of the classification rate. As a result, a considerably lower number of prototypes than with the initial algorithm is determined, while virtually preserving the classification rate (the degradation is minimal) or even improving it. Experiments also show that the new version of the heuristic leads, in general, to more efficient approaches than the original one.

6.1.3.2 Feature Selection

The aforementioned heuristic has also been incorporated to feature selection by means of a sequential forward generation procedure, thus allowing for a significant reduction in the number of texture methods to be evaluated during feature extraction and in the number of features to be evaluated during classification. A similar algorithm as with

UNIVERSITAT ROVIRA I VIRGILI
SUPERVISED AND UNSUPERVISED SEGMENTATION OF TEXTURED IMAGES BY EFFICIENT MULTI-LEVEL PATTERN CLASSIFICATION
Jaime Christian Meléndez Rodríguez
ISBN:978-84-693-7671-3/DL:T-1750-2010

prototype reduction is followed, but measuring the cost as the number of texture features instead of as the number of prototypes. As with the previous case, experiments have shown that the new version of the heuristic leads to a lower number of selected features than the original one, with very small degradation in the classification rate.

6.1.4 Support Vector Reduction

By utilizing the simplest linear kernels with the SVM-based classifier, a method for summarizing the several SVs resulting from SVM optimization into a single element has been incorporated. Experiments carried out in this thesis have demonstrated the suitability of the linear kernels for the texture classification task at hand and, furthermore, have demonstrated that, by sacrificing the small, extra percentage in classification rate that could be obtained with other kernels, a huge saving in computation is achieved.

6.1.5 Extension of the Supervised Methodology to the Unsupervised Domain

An inherent problem of current approaches for unsupervised texture segmentation is the lack of prior knowledge about the texture patterns to be discriminated, since, by definition, such information is not available. In this sense, a new methodology for unsupervised texture segmentation that incorporates the advantages of supervision has been proposed in this dissertation by extending the key aspects of the previously proposed methodology for supervised pixel-based classification. In order to accomplish this extension, the concept of pattern discovery has been introduced. Moreover, a segmentation quality measure derived from the classification rate used for evaluating supervised classifiers has also been developed.

6.1.5.1 Pattern Discovery

As its name suggests, the pattern discovery stage of the proposed methodology aims at determining the texture patterns present in a given image in order to provide a set of texture models that can be used later as input for a subsequent stage, such as the pixel-based classification stage utilized in this thesis, which is responsible for the final segmentation.

In this dissertation, three general purpose clustering algorithms have been utilized for pattern discovery. In order to overcome some of their limitations and also to achieve an efficient solution, several modifications to the baseline algorithms have been introduced. Moreover, it has been shown that the resulting algorithms do not need to obtain a highly accurate initial segmentation for the complete segmentation technique to succeed. Thus, only a reduced number of feature vectors are processed, which contributes to significantly reducing the computational burden.

6.1.5.2 Supervised Pixel-Based Texture Classification for Unsupervised Texture Segmentation

Once a set of texture models has been obtained, thus effectively transforming the original unsupervised problem into a supervised one, the application of the methodology for supervised pixel-based texture classification discussed so far is straightforward. At this point, the main concern is to choose the technique with the lowest associated cost both in training and in classification among the developed ones, which corresponds to the SVM-based classifier.

Experiments have shown that the proposed methodology outperforms many state-of-the-art segmenters in terms of segmentation quality, robustness and processing time, and is also better in terms of segmentation quality than its fully supervised counterpart.

UNIVERSITAT ROVIRA I VIRGILI
SUPERVISED AND UNSUPERVISED SEGMENTATION OF TEXTURED IMAGES BY EFFICIENT MULTI-LEVEL PATTERN CLASSIFICATION
Jaime Christian Meléndez Rodríguez
ISBN:978-84-693-7671-3/DL:T-1750-2010

6.1.5.3 Segmentation Quality Measure

The proposed segmentation quality measure is inspired by the classification rate utilized for assessing the performance of supervised pixel-based classifiers, but, due to the lack of correspondence between the labels in a segmentation map and the labels in a ground-truth, it compares and associates regions instead of pixels. By ensuring the uniqueness of these associations, both oversegmentation and undersegmentation are penalized.

6.2 Future Work

The methodologies described in this dissertation may be further extended in order to include new capabilities or give support to new application fields. Some of the future research lines are briefly outlined below.

6.2.1 Evaluation of Different Texture Features

This dissertation has shown that, although the used Gabor features yield the best segmentation/classification results in general terms, there are particular instances for which other texture features, such as LBP, lead to more accurate representations of texture. Therefore, it would be beneficial to incorporate those features into the proposed methodology. Moreover, in the supervised case, the developed feature selection procedure can be applied in order to determine the best set of features for every given problem.

6.2.2 Improve Feature Selection and Prototype Reduction for Supervised Texture Segmentation

The feature selection algorithm proposed in this thesis is based on a benefit-cost ratio analysis, where the benefit is measured in terms of the classification rate and

the cost is measured in terms of the number of evaluated features. Both measures could be improved by integrating alternative criteria. For instance, the separability between classes produced by a set of features or the degree of complementarity in classification of a given feature with respect to the previously analyzed ones could be added to the classification rate in order to measure the benefit. This could also be extended to prototype reduction by considering prototypes instead of features. On the other hand, if, as previously proposed, different families of texture methods are evaluated, the cost could be more precisely measured by considering the computation time of each feature and the method associated with it instead of by assigning the same cost to every feature.

6.2.3 Feature Selection for Unsupervised Texture Segmentation

Since the techniques for unsupervised texture segmentation proposed in this thesis have a supervised component, any feature selection algorithm oriented to supervised classification would be applicable. However, the main issue is the computation time associated with that algorithm, as it must be run online and, thus, accounts for the overall segmentation time. Therefore, a fast algorithm would be necessary.

According to the reviewed literature, an alternative may be to use a filter algorithm instead of a wrapper (such as the one developed in this thesis). Unfortunately, wrappers usually yield better solutions than filters, as the former are tailored to the classifier that later uses the resulting feature set. Furthermore, even when filters tend to be much faster than wrappers, there is no guarantee that an acceptable tradeoff between segmentation quality and processing time can be achieved.

Under such premises, a second alternative would be to devise some kind of offline feature selection procedure, which may seem impossible due to the lack of training models. However, by utilizing a set of general, representative texture models, such

as those based on perceptual features determined by (Rao and Lohse, 1996), a "one-time", application-independent feature selection algorithm, such as the one proposed in (Puig et al., 2010), can be formulated.

6.2.4 Minimum Evaluation Window Size for Supervised Texture Segmentation

It has been stated that, for texture segmentation, large evaluation windows are preferred inside regions of homogeneous texture, whereas small windows are preferred near boundaries between those regions in order to precisely locate the boundaries. The latter, however, is not completely true, as, usually, small evaluation windows have low discrimination power due to their reduced spatial coverage, which limits their capabilities of capturing enough texture information. As a consequence, classification with too small evaluation windows may lead to errors and to worsening boundary localization instead of improving it. Therefore, a procedure for determining the minimum acceptable window size, maybe by following a similar approach as the one defined in this dissertation for the maximum evaluation window size utilized for supervised texture segmentation, is desirable.

6.2.5 Minimum and Maximum Evaluation Window Sizes for Unsupervised Texture Segmentation

The issues regarding the minimum and maximum evaluation window sizes discussed above are also found in the unsupervised domain, although with the added difficulty of lacking a training set, which means that any method based on classifying the training set is unfeasible. In this case, those parameters must be determined directly from the image contents, perhaps by frequency or structural analysis.

6.2.6 Texture Segmentation and Classification of Color Images

The schemes developed in this work are oriented to gray-scale texture segmentation and classification. However, the spread of new technologies and applications requires the development of new algorithms for processing and analyzing color images. In this case, the key point is how to integrate both cues in a coherent way, as they do not necessarily complement each other and, furthermore, may yield contradictory segmentations when evaluated on their own.

6.3 Publications

The following publications have been derived from this thesis:

1. Melendez, J., Puig, D., Garcia, M. A., 2010. Multi-level pixel-based texture classification through efficient prototype selection via normalized cut. Pattern Recognition 43 (12), 4113–4123.

2. Melendez, J., Garcia, M. A., Puig, D., Petrou, M., 2010. Unsupervised texture-based image segmentation through pattern discovery. Submitted to Computer Vision and Image Understanding.

3. Melendez, J., Garcia, M. A., Puig, D., 2008. Efficient distance-based per-pixel texture classification with Gabor wavelet filters. Pattern Analysis and Applications 11(3-4), 365–372.

4. Puig, D., Melendez, J., Garcia, M. A., 2009. Texture-based approach for computer vision systems in autonomous vehicles. In: Solanas, A., Martínez-Ballesté, A. (Eds.), Intelligent Information Systems. Vol. 1, pp. 223–248.

5. Melendez, J., Puig, D., Garcia, M. A., 2010. On adapting pixel-based classification to unsupervised texture segmentation. In: Proceedings of the International Conference on Pattern Recognition. pp. 854–857.

6. Melendez, J., Puig, D., Garcia, M. A., 2009. Gabor-based texture classification through efficient prototype selection via normalized cut. In: Proceedings of the IEEE International Conference on Image Processing. pp. 1385–1388.

7. Melendez, J., Garcia, M. A., Puig, D., 2007. Supervised pixel-based texture classification with Gabor wavelet filters. In: Proceedings of the International Conference on Computer Vision Systems. <http://dx.doi.org/10.2390/biecoll-icvs2007-118>.

Other publications related to this thesis are:

1. Puig, D., Garcia, M. A., Melendez, J., 2010. Application-independent feature selection for texture classification. Pattern Recognition 43 (10), 3282–3297.

2. Melendez, J., Puig, D., Garcia, M. A., 2007. Comparative evaluation of classical methods, optimized Gabor filters and LBP for texture feature selection and classification. In: Proceedings of the International Conference on Computer Analysis of Images and Patterns, Lecture Notes in Computer Science. Vol. 4673. pp. 912–920.

3. Melendez, J., Garcia, M. A., Puig, D., 2006. Comparison of Local Binary Patterns, Gabor filters and integration of multiple methods for pixel-based texture classification. In: Segundas Jornadas de Investigación en Automática, Visión y Robótica. pp. 195–199.

UNIVERSITAT ROVIRA I VIRGILI
SUPERVISED AND UNSUPERVISED SEGMENTATION OF TEXTURED IMAGES BY EFFICIENT MULTI-LEVEL PATTERN CLASSIFICATION
Jaime Christian Meléndez Rodríguez
ISBN:978-84-693-7671-3/DL:T-1750-2010

UNIVERSITAT ROVIRA I VIRGILI
SUPERVISED AND UNSUPERVISED SEGMENTATION OF TEXTURED IMAGES BY EFFICIENT MULTI-LEVEL PATTERN CLASSIFICATION
Jaime Christian Meléndez Rodríguez
ISBN:978-84-693-7671-3/DL:T-1750-2010

Appendix A

Images Used in Experimentation

A broad collection of images of 256×256 pixels has been considered in this work. They correspond to the following sets: a) composite images of Brodatz (1999) and MeasTex (Smith and Burns, 1997) textures, b) composite images of Vistex textures (MIT Vision and Modeling Group, 1998), c) real outdoor images taken at ground level, d) real outdoor images taken by aerial devices and e) natural images from the Berkeley database (Martin et al., 2001). Figures A.1 to A.9 show these images. Both gray-scale and color versions (when available) are shown.

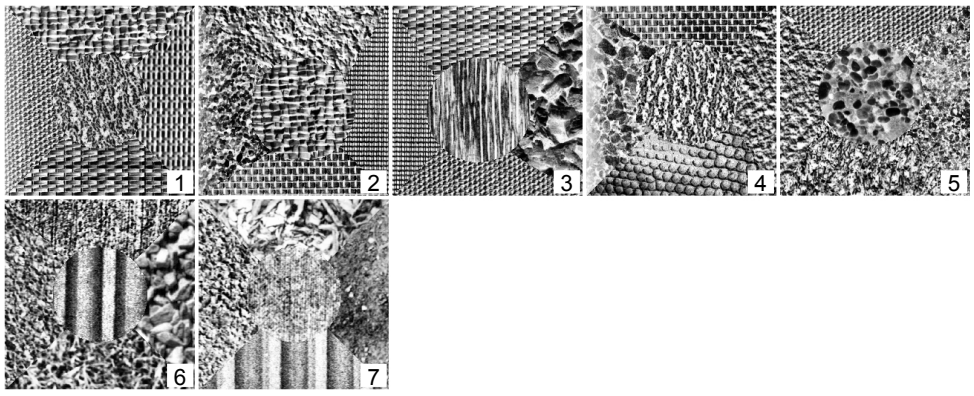

Figure A.1: Examples of Brodatz (top) and MeasTex (bottom) compositions.

UNIVERSITAT ROVIRA I VIRGILI
SUPERVISED AND UNSUPERVISED SEGMENTATION OF TEXTURED IMAGES BY EFFICIENT MULTI-LEVEL PATTERN CLASSIFICATION
Jaime Christian Meléndez Rodríguez
ISBN:978-84-693-7671-3/DL:T-1750-2010

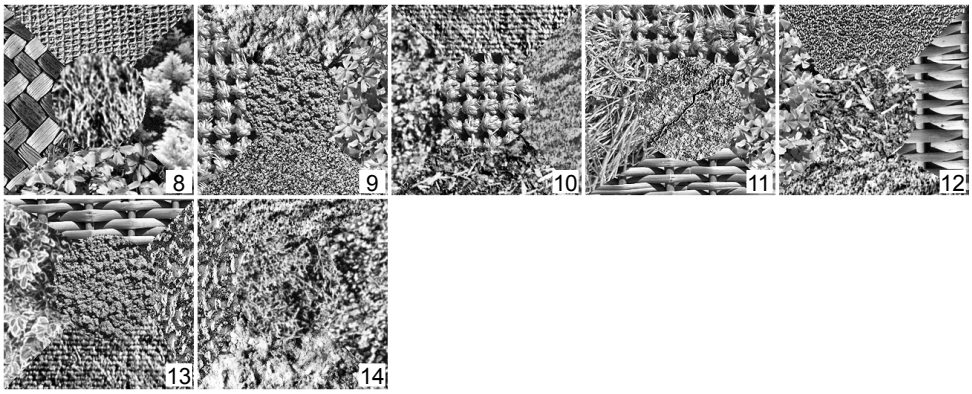

Figure A.2: Examples of VisTex compositions (gray-scale).

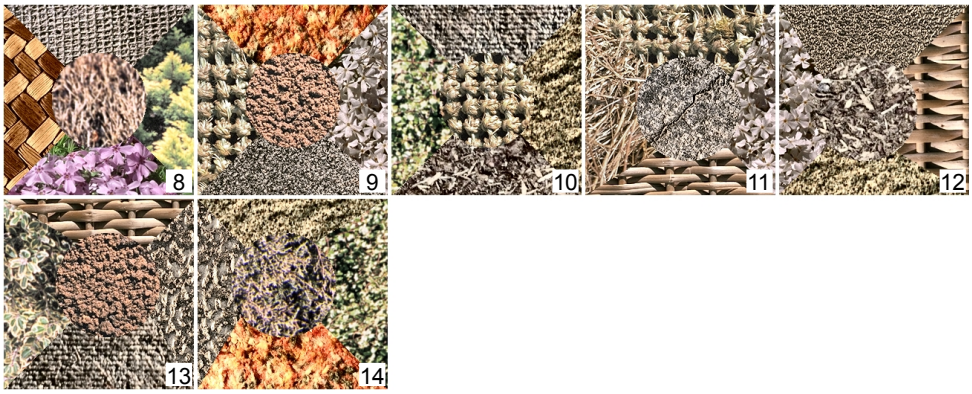

Figure A.3: Examples of VisTex compositions (color).

Figure A.4: Examples of outdoor images taken at ground level (gray-scale).

UNIVERSITAT ROVIRA I VIRGILI
SUPERVISED AND UNSUPERVISED SEGMENTATION OF TEXTURED IMAGES BY EFFICIENT MULTI-LEVEL PATTERN CLASSIFICATION
Jaime Christian Meléndez Rodríguez
ISBN:978-84-693-7671-3/DL:T-1750-2010

Figure A.5: Examples of outdoor images taken at ground level (color).

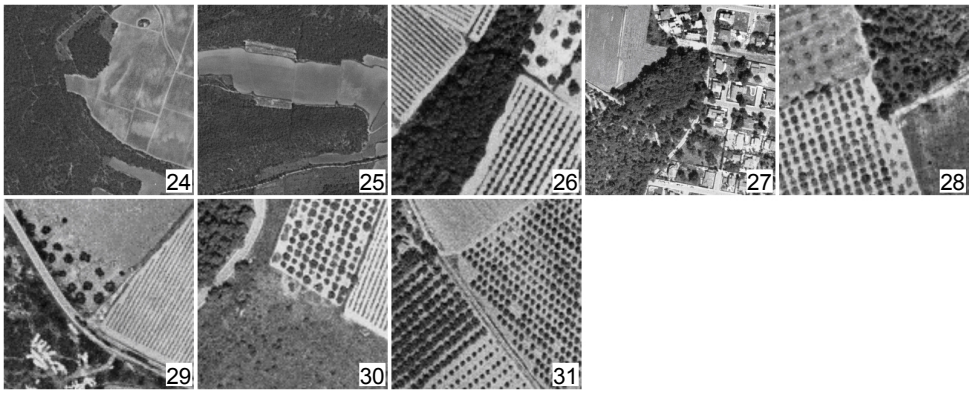

Figure A.6: Examples of outdoor images taken by aerial devices (gray-scale).

Figure A.7: Examples of outdoor images taken by aerial devices (color).

UNIVERSITAT ROVIRA I VIRGILI
SUPERVISED AND UNSUPERVISED SEGMENTATION OF TEXTURED IMAGES BY EFFICIENT MULTI-LEVEL PATTERN CLASSIFICATION
Jaime Christian Meléndez Rodríguez
ISBN:978-84-693-7671-3/DL:T-1750-2010

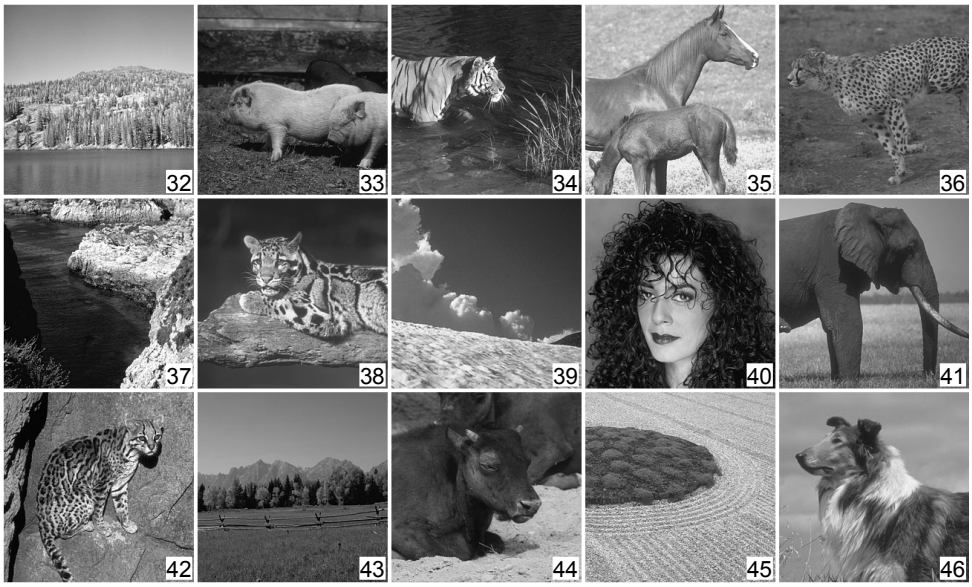

Figure A.8: Examples of natural images from the Berkeley database (gray-scale).

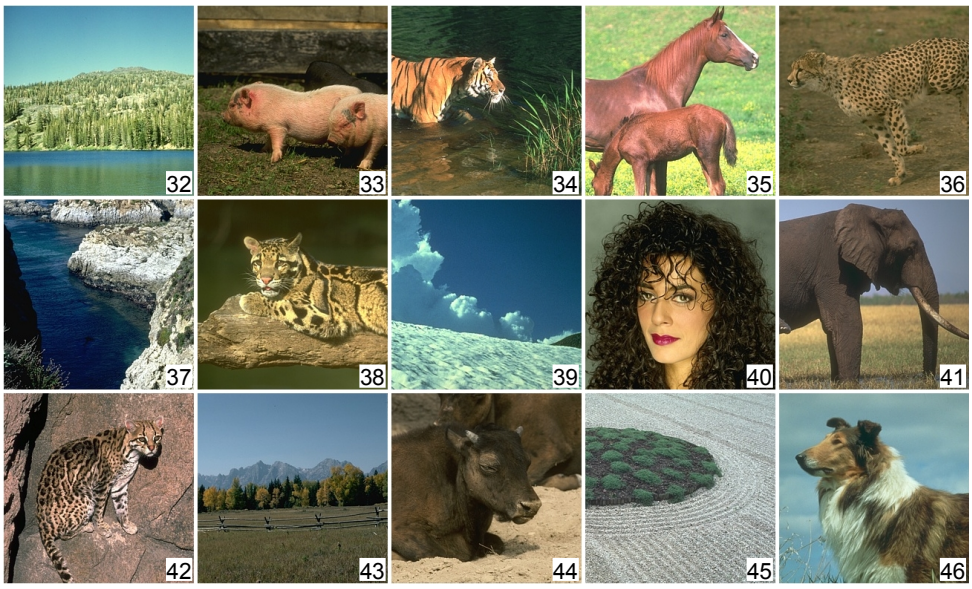

Figure A.9: Examples of natural images from the Berkeley database (color).

Bibliography

Abbadeni, N., Ziou, D., Wang, S., 2000. Computational measures corresponding to perceptual texture features. In: Proceedings of the IEEE International Conference on Image Processing. pp. 897–900.

Acharyya, M., De, R. K., Kundu, M. K., 2003. Extraction of features using m-band wavelet packet frame and their neuro-fuzzy evaluation for multitexture segmentation. IEEE Transactions on Pattern Analysis and Machine Intelligence 25 (12), 1639–1644.

Ahuja, N., Rosenfeld, A., 1981. Mosaic models for textures. IEEE Transactions on Pattern Analysis and Machine Intelligence PAMI-3 (1), 1–11.

Al-Janobi, A., 2001. Performance evaluation of cross-diagonal texture matrix method of texture analysis. Pattern Recognition 34 (1), 171–180.

Alemán-Flores, M., Alvarez, L., Caselles, V., 2007. Texture-oriented anisotropic filtering and geodesic active contours in breast tumor ultrasound segmentation. Journal of Mathematical Image and Vision 28 (1), 81–97.

Allili, M. S., Ziou, D., 2007. Globally adaptive region information for automatic color-texture image segmentation. Pattern Recognition Letters 28 (15), 1946–1956.

Almuallim, H., Dietterich, T. G., 1994. Learning boolean concepts in the presence of many irrelevant features. Artificial Intelligence 69 (1-2), 279–305.

Amadasun, M., King, R., 1989. Textural features corresponding to textural properties. IEEE Transactions on Systems, Man, and Cybernetics 19 (5), 1264–1274.

Arbeláez, P., Maire, M., Fowlkes, C., Malik, J., 2009. From contours to regions: an empirical evaluation. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 2294–2301.

Baheerathan, S., Albregsten, F., Danielsen, H. E., 1999. New texture features based on the complexity curve. Pattern Recognition 32 (4), 605–618.

Bajcsy, R., 1973. Computer description of textured surfaces. In: Proceedings of the International Joint Conferences on Artificial Intelligence. pp. 572–579.

Baltzakis, H., Papamarkos, N., 2001. A new signature verification technique based on a two-stage neural network classifier. Engineering Applications of Artificial Intelligence 14 (1), 95–103.

Ben-Bassat, M., 1982. Use of distance measures, information measures and error bounds in feature evaluation. In: Krishnaiah, P. R., Kanal, L. N. (Eds.), Handbook of Statistics. North-Holland Publishing Company, pp. 207–248.

Benboudjema, D., Pieczynski, W., 2007. Unsupervised statistical segmentation of nonstationary images using triplet Markov fields. IEEE Transactions on Pattern Analysis and Machine Intelligence 29 (8), 1367–1378.

Bennett, J., Khotanzad, A., 1998. Multispectral random field models for synthesis and analysis of color images. IEEE Transactions on Pattern Analysis and Machine Intelligence 20 (3), 327–332.

Bennett, J., Khotanzad, A., 1999. Maximum likelihood estimation methods for multispectral random field image models. IEEE Transactions on Pattern Analysis and Machine Intelligence 21 (6), 537–543.

Bianconi, F., Fernández, A., 2007. Evaluation of the effects of Gabor filter parameters on texture classification. Pattern Recognition 40 (12), 3325–3335.

Bishop, C. M., 2006. Pattern Recognition and Machine Learning. Springer.

Blostein, D., Ahuja, N., 1989. Shape from texture: integrating texture-element extraction and surface estimation. IEEE Transactions on Pattern Analysis and Machine Intelligence 11 (12), 1233–1251.

Boldys, J., 2003. Bayesian supervised segmentation of objects in natural images using low-level information. In: Proceedings of the International Symposium on Image and Signal Processing and Analysis. Vol. 2. pp. 1054–1059.

Bouman, C., Liu, B., 1991. Multiple resolution segmentation of textured images. IEEE Transactions on Pattern Analysis and Machine Intelligence 13 (2), 99–113.

Bresch, M., 2002. Optimal filter banks for supervised texture recognition. Pattern Recognition 35 (4), 783–790.

Brodatz, P., 1999. Textures: A Photographic Album for Artists and Designers. Dover & Gree Publishing Company.

Camilleri, K. P., Petrou, M., 2000. Spectral unmixing of mixed pixels for texture boundary refinement. In: Proceedings of the International Conference on Pattern Recognition. pp. 1096–1099.

Candès, E. J., Donoho, D. L., 2000. Curvelets – a surprisingly effective nonadaptive representation for objects with edges. In: Cohen, A., Rabut, C., Schumaker, L. L. (Eds.), Curve and Surface Fitting. Vanderbilt University Press, pp. 105–120.

Carlucci, L., 1972. A formal system for texture languages. Pattern Recognition 4 (1), 53–72.

Chakraborty, A., Duncan, J. S., 1999. Game-theoretic integration for image segmentation. IEEE Transactions on Pattern Analysis and Machine Intelligence 21 (1), 12–30.

Chan, T. F., Vese, L. A., 2001. Active contours without edges. IEEE Transactions on Image Processing 10 (2), 266–277.

Chang, T., Kuo, J., 1993. Texture analysis and classification with tree-structured wavelet transform. IEEE Transactions on Image Processing 2 (4), 429–441.

Chang, Y.-L., Li, X., 1994. Adaptive image region-growing. IEEE Transactions on Image Processing 3 (6), 868–872.

Charalampidis, D., Kasparis, T., 2002. Wavelet-based rotational invariant roughness features for texture classification and segmentation. IEEE Transactions on Image Processing 11 (8), 825–837.

Charalampidis, D., Kasparis, T., Georgiopoulos, M., 2001. Classification of noisy signals using fuzzy artmap neural networks. IEEE Transactions on Neural Networks 12 (5), 1023–1036.

Chaudhuri, B. B., Sarkar, N., 1995. Texture segmentation using fractal dimension. IEEE Transactions on Pattern Analysis and Machine Intelligence 17 (1), 72–77.

Chellappa, R., Chatterjee, S., 1985. Classification of textures using Gaussian Markov fields. IEEE Transactions on Acoustics, Speech and Signal Processing ASSP-33 (4), 959–963.

Chen, J. L., 1997. A simplified approach to the HMM based texture analysis and its application to document segmentation. Pattern Recognition Letters 18 (10), 993–1007.

Chen, J. L., Kundu, A., 1994. Rotation and gray-scale transform invariant texture identification using wavelet decomposition and hidden Markov model. IEEE Transactions on Pattern Analysis and Machine Intelligence 16 (2), 208–214.

Chen, L., Lu, G., Zhang, D., 2004. Effects of different Gabor filters parameters on image retrieval by texture. In: Proceedings of the International Multimedia Modeling Conference. pp. 273–278.

Chen, Y. Q., Nixon, M. S., Thomas, D. W., 1995. Statistical geometrical features for texture classification. Pattern Recognition 28 (4), 537–552.

Chitre, Y., Dhawan, A. P., 1999. M-band wavelet discrimination of natural textures. Pattern Recognition 32 (5), 773–789.

Choi, H., Baraniuk, R. G., 2001. Multiscale image segmentation using wavelet-domain hidden Markov models. IEEE Transactions on Image Processing 10 (9), 1309–1321.

Choy, S. K., Tong, C. S., 2008. Statistical properties of bit-plane probability model and its application in supervised texture classification. IEEE Transactions on Image Processing 17 (8), 1399–1405.

Clark, M., Bovik, A. C., 1986. Texture discrimination using a model of visual cortex. In: Proceedings of the IEEE International Conference on Systems, Man, and Cybernetics.

Clerc, M., Mallat, S., 2002. The texture gradient equation for recovering shape from texture. IEEE Transactions on Pattern Analysis and Machine Intelligence 24 (4), 536–549.

Coetzee, F. M., 2005. Correcting the Kullback-Leibler distance for feature selection. Pattern Recognition Letters 26 (11), 1675–1683.

Comaniciu, D., Meer, P., 2002. Mean shift: a robust approach toward feature space analysis. IEEE Transactions on Pattern Analysis and Machine Intelligence 24 (5), 603–619.

Comer, M. L., Delp, E. J., 1999. Segmentation of textured images using a multiresolution Gaussian autoregressive model. IEEE Transactions on Image Processing 8 (3), 408–420.

Crammer, K., Singer, Y., 2002. On the learnability and design of output codes for multiclass problems. Machine Learning 47 (2-3), 201–233.

Cross, G. C., Jain, A. K., 1983. Markov random field texture models. IEEE Transactions on Pattern Analysis and Machine Intelligence 5 (1), 25–39.

Cui, P., Li, J., Pan, Q., Zhang, H., 2006. Rotation and scaling invariant texture classification based on Radon transform and multiscale analysis. Pattern Recognition Letters 27 (5), 408–413.

Cula, O. G., Dana, K. J., 2001. Compact representation of bidirectional texture functions. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. Vol. 1. pp. 1041–1047.

Dasgupta, N., Carin, L., 2006. Texture analysis with variational hidden Markov trees. IEEE Transactions on Signal Processing 54 (6), 2353–2356.

Dash, M., Liu, H., 1997. Feature selection for classification. Intelligent Data Analysis 1 (3), 131–156.

Daugman, J. G., 1985. Uncertainty relation for resolution in space, spatial frequency, and orientation optimized by two-dimensional visual cortical filters. Journal of the Optical Society of America A 2 (7), 1160–1169.

Davidson, J. L., Cressie, N., Hua, X., 1999. Texture synthesis and pattern recognition for partially ordered Markov models. Pattern Recognition 32 (9), 1475–1505.

Deng, H., Clausi, D. A., 2004. Unsupervised image segmentation using a simple MRF model with a new implementation scheme. Pattern Recognition 37 (12), 2323–2335.

Deng, Y., Manjunath, B. S., 2001. Unsupervised segmentation of color-texture regions in images and video. IEEE Transactions on Pattern Analysis and Machine Intelligence 23 (8), 800–810.

Derin, H., Elliott, H., Cristi, R., Geman, D., 1984. Bayes smoothing algorithms for segmentation of binary images modeled by Markov random fields. IEEE Transactions on Pattern Analysis and Machine Intelligence 6 (6), 707–720.

Dhillon, I. S., Guan, Y., Kulis, B., 2007. Weighted graph cuts without eigenvectors: a multilevel approach. IEEE Transactions on Pattern Analysis and Machine Intelligence 29 (11), 1944–1957.

Do, M. N., Vetterli, M., 2002. Rotation invariant texture characterization and retrieval using steerable wavelet-domain hidden Markov models. IEEE Transactions on Multimedia 4 (4), 517–527.

Doak, J., 1992. An evaluation of feature selection methods and their application to computer security. Tech. rep., University of California at Davis, Dept. Computer Science.

Dokur, Z., Ölmez, T., 2002. Segmentation of ultrasound images by using a hybrid neural network. Pattern Recognition Letters 23 (14), 1825–1836.

Dondes, P. A., Rosenfeld, A., 1982. Pixel classification based on gray level and local "busyness". IEEE Transactions on Pattern Analysis and Machine Intelligence 4 (1), 79–84.

Dong, X., Pollak, I., 2006. Multiscale segmentation with vector-valued nonlinear diffusions on arbitrary graphs. IEEE Transactions on Image Processing 15 (7), 1993–2005.

Du, G., Yeo, T. S., 2002. A novel lacunarity estimation method applied to SAR image segmentation. IEEE Transactions on Geoscience and Remote Sensing 40 (12), 2687–2691.

Duda, R. O., Hart, P. E., Stork, D. G., 2001. Pattern Classification. John Wiley & Sons.

Dunn, D., Higgins, W. E., 1995. Optimal Gabor filters for texture segmentation. IEEE Transactions on Image Processing 4 (7), 947–964.

Dyer, C. R., Rosenfeld, A., 1976. Fourier texture features: suppression of aperture effects. IEEE Transactions on Systems, Man, and Cybernetics SMC-6 (10), 703–705.

Efron, B., Tibshirani, R., Storey, J. D., Tusher, V., 2001. Empirical Bayes analysis of a microarray experiment. Journal of the American Statistical Association 96 (456), 106–120.

Fan, G., Xia, X.-G., 2003. Wavelet-based texture analysis and synthesis using hidden Markov models. IEEE Transactions on Circuits and Systems Part I 50 (1), 106–120.

Farid, H., Kosecka, J., 2007. Estimating planar surface orientation using bispectral analysis. IEEE Transactions on Image Processing 16 (8), 2154–2160.

Fleuret, F., 2004. Fast binary feature selection with conditional mutual information. Journal of Machine Learning Research 5 (1531–1555).

Forman, G., 2003. An extensive empirical study of feature selection metrics for text classification. Journal of Machine Learning Research 4 (1289–1305).

Fox, R. J., Dimmic, M. W., 2006. A two-sample Bayesian t-test for microarray data. BMC Bioinformatics 7 (126).

Freeman, W. T., Adelson, E. H., 1991. The design and use of steerable filters. IEEE Transactions on Pattern Analysis and Machine Intelligence 19 (9), 891–906.

Freixenet, J., Muñoz, X., Raba, D., Martí, J., Cufí, X., 2002. Yet another survey on image segmentation: region and boundary information integration. In: Proceedings of the European Conference on Computer Vision, Lecture Notes in Computer Science. Vol. 2352. pp. 408–422.

Gabor, D., 1946. Theory of communication. Journal of the Institution of Electrical Engineers 93, 429–457.

Galloway, M. M., 1975. Texture analysis using gray level run lengths. Computer Graphics and Image Processing 4 (2), 172–179.

Gangepain, J. J., Roques-Carmes, C., 1986. Fractal approach to two-dimensional and three-dimensional surface roughness. Wear 109 (1-4), 119–126.

Garcia, M. A., Puig, D., 2003. Pixel classification by divergence-based integration of multiple texture methods and its application to fabric defect detection. In: Proceedings of the DAGM-Symposium, Lecture Notes in Computer Science. pp. 132–139.

Garcia-Sevilla, P., Petrou, M., 2000. Analysis of irregularly shaped texture regions: a comparative study. In: Proceedings of the International Conference on Pattern Recognition. pp. 1080–1083.

Gelzinis, A., Verikas, A., Bacauskiene, M., 2007. Increasing the discrimination power of the co-occurrence matrix-based features. Pattern Recognition 40 (9), 2367–2372.

Georgescu, B., Shimshoni, I., Meer, P., 2003. Mean shift based clustering in high dimensions: a texture classification example. In: Proceedings of the IEEE International Conference on Computer Vision. Vol. 1. pp. 456–463.

Guyon, I., Elisseeff, A., 2003. An introduction to variable and feature selection. Journal of Machine Learning Research 3, 1157–1182.

Hall, M., 1999. Correlation-Based Feature Selection for Machine Learning. Ph.D. thesis, Department of Computer Science, Waikato University, New Zealand.

Hamerly, G., Elkan, C., 2003. Learning the k in k-means. In: Thrun, S., Saul, L., Schölkopf, B. (Eds.), Advances in Neural Information Processing Systems. pp. 281–288.

Haralick, R. M., 1979. Statistical and structural approaches to texture. Proceedings of the IEEE 67 (5), 786–804.

Haralick, R. M., Shanmugam, K., Dinstein, I., 1973. Textural features for image classification. IEEE Transactions on Systems, Man, and Cybernetics SMC-3 (6), 610–621.

Haralick, R. M., Shapiro, L. G., 1992. Computer and Robot Vision. Vol. 1. Addison-Wesley Longman Publishing Co.

Hatipoglu, S., Mitra, S. K., Kingsbury, N., 2000. Image texture description using complex wavelet transform. In: Proceedings of the IEEE International Conference on Image Processing. pp. 530–533.

He, D.-C., Wang, L., 1990. A new statistical approach to texture analysis. Photogrammetric Engineering and Remote Sensing 56 (1), 61–66.

Heaps, C., Handel, S., 1999. Similarity and features of natural textures. Journal of Experimental Psychology: Human Perception and Performance 25 (2), 199–320.

Hernández, B., Olague, G., Hammoud, R., Trujillo, L., Romero, E., 2007. Visual learning of texture descriptors for facial expression recognition in thermal imagery. Computer Vision and Image Understanding 106 (2-3), 258–269.

Horikawa, Y., 2004. Comparison of support vector machines with autocorrelation kernels for invariant texture classification. In: Proceedings of the International Conference on Pattern Recognition. pp. 660–663.

Hsin, H.-C., Li, C.-C., 1998. An experiment on texture segmentation using modulated wavelets. IEEE Transactions on Systems, Man, and Cybernetics–Part A 28 (5), 720–725.

Hsu, C.-W., Lin, C.-J., 2002. A comparison of methods for multiclass support vector machines. IEEE Transactions on Neural Networks 13 (2), 415–425.

Hsu, T.-I., Kuo, J. L., Wilson, R., 2000. A multiresolution texture gradient method for unsupervised segmentation. Pattern Recognition 33 (1), 1819–1833.

Huang, P. W., Dai, S. K., Lin, P. L., 2006. Texture image retrieval and image segmentation using composite sub-band gradient vectors. Journal of Visual Communication and Image Representation 17 (5), 947–957.

Ilow, J., Leung, H., 2001. Self-similar texture modeling using FARIMA processes with applications to satellite images. IEEE Transactions on Image Processing 10 (5), 792–797.

Inza, I., Larrañaga, P., Blanco, R., Cerrolaza, A. J., 2004. Filter versus wrapper gene selection approaches in DNA microarray domains. Artificial Intelligence in Medicine 31 (2), 91–103.

Ivins, J., Porrill, J., 1995. Active region models for segmenting textures and colours. Image and Vision Computing 13 (5), 431–438.

Jain, A. K., Duin, R. P. W., Mao, J., 2000. Statistical pattern recognition: a review. IEEE Transactions on Pattern Analysis and Machine Intelligence 22 (1), 4–7.

Jain, A. K., Karu, K., 1996. Learning texture discrimination masks. IEEE Transactions on Pattern Analysis and Machine Intelligence 18 (2), 195–205.

Jalba, A. C., Wilkinson, M. H. F., Roerdink, J. B. T. M., 2004. Morphological hat-transform scale spaces and their use in pattern classification. Pattern Recognition 37 (5), 901–915.

Jayaramamurthy, S. N., 1979. Multilevel array grammars for generating texture scenes. In: Proceedings of the IEEE Conference on Pattern Recognition and Image Processing. pp. 391–398.

John, G. H., Kohavi, R., Pfleger, K., 1994. Irrelevant features and the subset selection problem. In: Proceedings of the International Conference on Machine Learning. pp. 121–129.

Julesz, B., 1975. Experiments in the visual perception of texture. Scientific American 232 (4), 34–43.

Julesz, B., 1981. Textons, the elements of texture perception, and their interactions. Nature 290 (5802), 91–97.

Julesz, B., 1986. Texton gradients: the texton theory revisited. Biological Cybernetics 54, 245–251.

Jung, S.-H., 2001. Content-based image retrieval using fuzzy multiple attribute relational graph. In: Proceedings of the IEEE International Symposium on Industrial Electronics. Vol. 3. pp. 1508–1513.

Jurie, F., Triggs, B., 2005. Creating efficient codebooks for visual recognition. In: Proceedings of the IEEE International Conference on Computer Vision. pp. 604–610.

Kalinin, M., Raicu, D. S., Furst, J. D., Channin, D. S., 2005. A classification approach for anatomical regions segmentation. In: Proceedings of the IEEE International Conference on Image Processing. Vol. 2. pp. 1262–1265.

Kameyama, K., Taga, K., 2004. Texture classification by support vector machines with kernels for higher-order Gabor filtering. In: Proceedings of the International Joint Conference on Neural Networks. Vol. 1. pp. 3009–3014.

Kaplan, L. M., Kuo, C.-C., 1995. Texture roughness analysis and synthesis via extended self-similar (ESS) model. IEEE Transactions on Pattern Analysis and Machine Intelligence 17 (11), 1043–1056.

Kashyap, R. L., 1984. Characterization and estimation of two-dimensional ARMA models. IEEE Transactions on Information Theory IT-30 (5), 736–745.

Kashyap, R. L., Khotanzad, A., 1986. A model-based method for rotation invariant texture classification. IEEE Transactions on Pattern Analysis and Machine Intelligence PAMI-8 (4), 472–481.

Keller, J., Crownover, R., Chen, S., 1989. Texture description and segmentation through fractal geometry. Computer Vision, Graphics, and Image Processing 45 (2), 150–160.

Khotanzad, A., Chen, J.-Y., 2002. Unsupervised segmentation of textured images by edge detection in multidimensional feature. IEEE Transactions on Pattern Analysis and Machine Intelligence 11 (4), 414–421.

Kim, J.-S., Hong, K.-S., 2009. Color-texture segmentation using unsupervised graph cuts. Pattern Recognition 42 (5), 735–750.

Kim, K. I., Jung, K., Park, S. H., Kim, H. J., 2002. Support vector machines for texture classification. IEEE Transactions on Pattern Analysis and Machine Intelligence 24 (11), 1542–1550.

Kim, N.-D., Udpa, S., 2000. Texture classification using rotated wavelet filters. IEEE Transactions on Systems, Man, and Cybernetics–Part A 30 (6), 847–852.

Kim, S. C., Kang, T. J., 2007. Texture classification and segmentation using wavelet packet frame and Gaussian mixture model. Pattern Recognition 40 (4), 1207–1221.

Koller, D., Sahami, M., 1996. Toward optimal feature selection. In: Proceedings of the International Conference on Machine Learning. pp. 284–292.

Kreßel, U. H.-G., 1999. Pairwise classification and support vector machines. In: Schölkopf, B., Burges, C. J. C., Smola, A. J. (Eds.), Advances in Kernel Methods: Support Vector Learning. MIT Press, pp. 255–268.

Krishnamachari, S., Chellappa, R., 1997. Multiresolution Gauss-Markov random field models for texture segmentation. IEEE Transactions on Image Processing 6 (2), 251–267.

Kumar, A., Zhang, D., 2006. Personal recognition using hand shape and texture. IEEE Transactions on Image Processing 15 (8), 2454–2461.

Kurmyshev, E. V., Sánchez-Yáñez, R. E., 2005. Comparative experiment with colour texture classifiers using the CCR feature space. Pattern Recognition Letters 26 (9), 1346–1353.

Laine, A., Fan, J., 1993. Texture classification by wavelet packet signatures. IEEE Transactions on Pattern Analysis and Machine Intelligence 15 (11), 1186–1191.

Laine, A., Fan, J., 1996. Frame representations for texture segmentation. IEEE Transactions on Image Processing 5 (5), 771–780.

Larlus, D., Jurie, F., 2009. Latent mixture vocabularies for object categorization and segmentation. Image and Vision Computing 27 (5), 523–534.

Laws, K. I., 1980. Textured Image Segmentation. Ph.D. thesis, Image Processing Institute, University of Southern California.

Lazebnik, S., Raginsky, M., 2009. Supervised learning of quantizer codebooks by information loss minimization. IEEE Transactions on Pattern Analysis and Machine Intelligence 31 (7), 1294–1309.

Lazebnik, S., Schmid, C., Ponce, J., 2005. A maximum entropy framework for part-based texture and object recognition. In: Proceedings of the IEEE International Conference on Computer Vision. Vol. 1. pp. 832–838.

Lee, K.-L., Chen, L. H., 2005. An efficient computation method for the texture browsing descriptor of MPEG-7. Image and Vision Computing 23 (5), 479–489.

Leung, T., Malik, J., 2001. Representing and recognizing the visual appearance of materials using three-dimensional textons. International Journal of Computer Vision 43 (1), 29–44.

Liao, S., Law, M. W. K., Chung, A. C. S., 2009. Dominant local binary patterns for texture classification. IEEE Transactions on Image Processing 18 (5), 1107–1118.

Lin, H.-T., Lin, C.-J., Weng, R. C., 2007. A note on Platt's probabilistic outputs for support vector machines. Machine Learning 38 (3), 267–276.

Liu, H., Motoda, H., 1998. Feature Selection for Knowledge Discovery and Data Mining. Kluwer Academic Publishers.

Liu, H., Setiono, R., 1996. A probabilistic approach to feature selection: a filter solution. In: Proceedings of the International Conference on Machine Learning. pp. 319–327.

Liu, H., Yu, L., 2005. Toward integrating feature selection algorithms for classification and clustering. IEEE Transactions on Knowledge and Data Engineering 17 (4), 491–502.

Liu, J.-C., Hwang, W.-L., Chen, M.-S., 2000. Estimation of 2-D noisy fractional Brownian motion and its application using wavelets. IEEE Transactions on Image Processing 9 (8), 1407–1419.

Liu, S.-C., Chang, S., 1997. Dimension estimation of discrete-time fractional Brownian motion with applications to image texture classification. IEEE Transactions on Image Processing 6 (8), 1176–1184.

Long, H., Leow, W. K., 2001. Perceptual texture space improves perceptual consistency of computational features. In: Proceedings of the International Joint Conferences on Artificial Intelligence. pp. 1391–1396.

Lu, S. Y., Fu, K. S., 1978. A syntactic approach for texture analysis. Computer Graphics and Image Processing 7 (3), 303–330.

Lumia, R., Haralick, R. M., Zuniga, O., Shapiro, L., Pong, T.-C., Wangand, F.-P., 1983. Texture analysis of aerial photographs. Pattern Recognition 16 (1), 39–46.

Lundahl, T., Ohley, W. J., Kay, S. M., Siffert, R., 1986. Fractional Brownian motion: a maximum likelihood estimator and its application to image texture. IEEE Transactions on Medical Imaging MI-5 (3), 152–161.

Luo, J., Savakis, A. E., 2000. Two-stage texture segmentation using complementary features. In: Proceedings of the IEEE International Conference on Image Processing. Vol. 3. pp. 564–567.

Ma, S., Huang, J., 2005. Regularized ROC method for disease classification and biomarker selection with microarray data. Bioinformatics 21 (24), 4356–4362.

Ma, W.-Y., Manjunath, B. S., 2000. Edgeflow: a technique for boundary detection and image segmentation. IEEE Transactions on Image Processing 9 (8), 1375–1388.

Maire, M., Arbeláez, P., Fowlkes, C., Malik, J., 2008. Using contours to detect and localize junctions in natural images. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 1–8.

Makeyev, O., Sazonov, E., Baidyk, T., Martín, A., 2008. Limited receptive area neural classifier for texture recognition of mechanically treated metal surfaces. Neurocomputing 71 (7-9), 1413–1421.

Malik, J., Belongie, S., Shi, J., Leung, T., 1999. Textons, contours and regions: cue integration in image segmentation. In: Proceedings of the IEEE International Conference on Computer Vision. Vol. 2. pp. 918–925.

Malik, J., Perona, P., 1990. Preattentive texture discrimination with early vision mechanisms. Journal of the Optical Society of America A 7 (5), 923–932.

Mallat, S. G., 1989. A theory for multiresolution signal decomposition: the wavelet representation. IEEE Transactions on Pattern Analysis and Machine Intelligence 11 (7), 674–693.

Mamitsuka, H., 2006. Selecting features in microarray classification using ROC curves. Pattern Recognition 39 (12), 2393–2404.

Manduchi, R., 1999. Bayesian fusion of color and texture segmentations. In: Proceedings of the IEEE International Conference on Computer Vision. Vol. 2. pp. 956–962.

Manjunath, B. S., Chellappa, R., 1993. A unified approach to boundary perception: edges, textures, and illusory contours. IEEE Transactions on Neural Networks 4 (1), 96–108.

Manjunath, B. S., Ma, W. Y., 1996. Texture features for browsing and image retrieval. IEEE Transactions on Pattern Analysis and Machine Intelligence 18 (8), 837–842.

Manjunath, B. S., Ohm, J.-R., Vasudevan, V. V., Yamada, A., 2001. Color and texture descriptors. IEEE Transactions on Circuits and Systems for Video Technology 11 (6), 703–715.

Mao, J., Jain, A. K., 1992. Texture classification and segmentation using multiresolution simultaneous autoregressive models. Pattern Recognition 25 (2), 173–188.

Mao, J., Jain, A. K., 1996. A self-organizing network for hyperellipsoidal clustering (HEC). IEEE Transactions on Neural Networks 7 (1), 16–29.

Marr, D., 1982. Vision. Freeman.

Martin, D., Fowlkes, C., Tal, D., Malik, J., 2001. A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics. In: Proceedings of the IEEE International Conference on Computer Vision. Vol. 2. pp. 416–423.

McCormick, B. H., Jayaramamurthy, S. N., 1974. Time series model for texture synthesis. International Journal of Computer and Information Sciences 3 (4), 329–343.

Melendez, J., Garcia, M. A., Puig, D., 2008. Efficient distance-based per-pixel texture classification with Gabor wavelet filters. Pattern Analysis and Applications 11 (3–4), 365–372.

Melendez, J., Puig, D., Garcia, M. A., 2007. Comparative evaluation of classical methods, optimized Gabor filters and LBP for texture feature selection and classification. In: Proceedings of the International Conference on Computer Analysis of Images and Patterns, Lecture Notes in Computer Science. Vol. 4673. pp. 912–920.

Mirmehdi, M., Petrou, M., 2000. Segmentation of color textures. IEEE Transactions on Pattern Analysis and Machine Intelligence 22 (2), 142–159.

MIT Vision and Modeling Group, 1998. <http://www.media.mit.edu/vismod>.

Mojsilović, A., Kovačević, J., Hu, J., Safranek, R. J., Ganapathy, S. K., 2000. Matching and retrieval based on the vocabulary and grammar of color patterns. IEEE Transactions on Image Processing 9 (1), 38–54.

Molina, L. C., Belanche, L., Nebot, A., 2002. Feature selection algorithms: a survey and experimental evaluation. Tech. rep., LSI-02-62-R, Universitat Politècnica de Catalunya.

Moosmann, F., Nowak, E., Jurie, F., 2008. Randomized clustering forests for image classification. IEEE Transactions on Pattern Analysis and Machine Intelligence 30 (9), 1632–1646.

Muñoz, X., Freixenet, J., Cufí, X., Martí, J., 2003. Active regions for clolour texture segmentation integrating region and boundary information. In: Proceedings of the IEEE International Conference on Image Processing. pp. 453–456.

Muneeswaran, K., Ganesan, L., Arumugam, S., Soundar, K. R., 2006. Texture image segmentation using combined features from spatial and spectral distribution. Pattern Recognition Letters 27 (7), 755–764.

Narendra, P. M., Fukunaga, K., 1977. A branch and bound algorithm for feature subset selection. IEEE Transactions on Computers 26 (9), 917–922.

Nister, D., Stewenius, H., 2006. Scalable recognition with a vocabulary tree. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 2161–2168.

Noda, H., Shirazi, M. N., Kawaguchi, E., 2002. MRF-based texture segmentation using wavelet decomposed images. Pattern Recognition 35 (4), 771–782.

Ojala, T., Pietikäinen, M., 1999. Unsupervised texture segmentation using feature distributions. Pattern Recognition 32 (3), 477–486.

Ojala, T., Pietikäinen, M., Harwood, D., 1996. A comparative study of texture measures with classification based on featured distributions. Pattern Recognition 29 (1), 51–59.

Ojala, T., Pietikäinen, M., Mäenpää, T., 2002. Multiresolution gray-scale and rotation invariant texture classification with local binary patterns. IEEE Transactions on Pattern Analysis and Machine Intelligence 24 (7), 971–987.

Ojala, T., Valkealahti, K., Oja, E., Pietikäinen, M., 2001. Texture discrimination with multidimensional distributions of signed gray-level differences. Pattern Recognition 34 (3), 727–739.

Omran, M., Engelbrecht, A., Salman, A., 2007. An overview of clustering methods. Intelligent Data Analysis 11 (6), 583–605.

Ozden, M., Polat, E., 2007. A color image segmentation approach for content-based image retrieval. Pattern Recognition 40 (4), 1318–1325.

Paget, R., 2004. Strong Markov random field. IEEE Transactions on Pattern Analysis and Machine Intelligence 26 (3), 408–413.

Pan, J. S., McInnes, F. R., Jack, M. A., 1996. Fast clustering algorithms for vector quantization. Pattern Recognition 29 (3), 511–518.

Papari, G., Petkov, N., 2009. Continuous glass patterns for painterly rendering. IEEE Transactions on Image Processing 18 (3), 652–664.

Paragios, N., Deriche, R., 2002. Geodesic active regions and level set methods for supervised texture segmentation. International Journal of Computer Vision 46 (3), 223–247.

Park, H.-A., Park, K. R., 2007. Iris recognition based on score level fusion by using SVM. Pattern Recognition Letters 28 (15), 2019–2028.

Peleg, S., Naor, J., Hartley, R., Avnir, D., 1984. Multiple resolution texture analysis and classification. IEEE Transactions on Pattern Analysis and Machine Intelligence PAMI-6 (4), 518–523.

Pelleg, D., Moore, A., 2000. X-means: extending k-means with efficient estimation of the number of clusters. In: Proceedings of the International Conference on Machine Learning. pp. 727–734.

Peng, H., Long, F., Ding, C., 2005. Feature selection based on mutual information: criteria of max-dependency, max-relevance, and min-redundancy. IEEE Transactions on Pattern Analysis and Machine Intelligence 27 (8), 1226–1238.

Pentland, A. P., 1984. Fractal-based description of natural scenes. IEEE Transactions on Pattern Analysis and Machine Intelligence PAMI-6 (6), 661–674.

Perona, P., Malik, J., 1990. Scale-space and edge detection using anisotropic diffusion. IEEE Transactions on Pattern Analysis and Machine Intelligence 12 (7), 629–639.

Perronnin, F., 2008. Universal and adapted vocabularies for generic visual categorization. IEEE Transactions on Pattern Analysis and Machine Intelligence 30 (7), 1243–1256.

Petrou, M., Garcia Sevilla, P., 2006. Image Processing: Dealing with Texture. John Wiley.

Peyré, G., 2010. Texture synthesis with grouplets. IEEE Transactions on Pattern Analysis and Machine Intelligence 32 (4), 733–664.

Pickett, R. M., 1970. Visual analysis of texture in the detection and recognition of objects. In: Lipkin, B. S., Rosenfeld, A. (Eds.), Picture Processing and Psychopictorics. Academic Press, pp. 298–308.

Platt, J. C., 2000. Probabilistic outputs for support vector machines and comparisons to regularized likelihood methods. In: Smola, A. J., Bartlett, P. L., Schölkopf, B., Schuurmans, D. (Eds.), Advances in Large Margin Classifiers. MIT Press, pp. 61–74.

Platt., J. C., Cristianini, N., Showe-Taylor, J., 2000. Large margin DAGs for multiclass classification. In: Solla, S. A., Leen, T. K., Müller, K.-R. (Eds.), Advances in Neural Information Processing Systems. MIT Press, pp. 547–553.

Po, D. D.-Y., Do, M. N., 2006. Directional multiscale modeling of images using the contourlet transform. IEEE Transactions on Image Processing 15 (6), 1610–1620.

Portnoff, M., 1980. Time-frequency representation of digital signals and systems based on short-time Fourier analysis. IEEE Transactions on Acoustics, Speech and Signal Processing 28 (1), 55–69.

Puig, D., Garcia, M. A., 2001. Determining optimal window size for texture feature extraction methods. In: Spanish Symposium on Pattern Recognition and Image Analysis. Vol. 2. pp. 237–242.

Puig, D., Garcia, M. A., 2003. Pixel-based texture classification by integration of multiple texture feature evaluation windows. In: Proceedings of the Iberian Conference on Pattern Recognition and Image Analysis, Lecture Notes in Computer Science. pp. 793–801.

Puig, D., Garcia, M. A., 2006. Automatic texture feature selection for image pixel classification. Pattern Recognition 39 (11), 1996–2009.

Puig, D., Garcia, M. A., Melendez, J., 2010. Application-independent feature selection for texture classification. Pattern Recognition 43 (10), 3282–3297.

Puzicha, J., Hofmann, T., Buhmann, J. M., 2000. A theory of proximity based clustering: structure detection by optimization. Pattern Recognition 33 (4), 617–634.

Raghu, P. P., Yegnanarayana, B., 1998. Supervised texture classification using a probabilistic neural network and constraint satisfaction model. IEEE Transactions on Neural Networks 9 (3), 516–522.

Rajpoot, K. M., Rajpoot, N. M., 2004. Wavelets and support vector machines for texture classification. In: Proceedings of the IEEE International Multitopic Conference. pp. 328–333.

Randen, T., Husøy, J. H., 1999. Filtering for texture classification: a comparative study. IEEE Transactions on Pattern Analysis and Machine Intelligence 21 (4), 291–310.

Rao, A. R., Lohse, G. L., 1993. Identifying high level features of texture perception. CVGIP: Graphical Models and Image Processing 55 (3), 218–233.

Rao, A. R., Lohse, G. L., 1996. Towards a texture naming system: identifying relevant dimensions of texture. Vision Research 36 (11), 1649–1669.

Rao, S., Mobahi, H., Yang, A., Sastry, S., 2009. Natural image segmentation with adaptive texture and boundary encoding. In: Proceedings of the Asian Conference on Computer Vision. Vol. 1. pp. 135–146.

Ray, N., Havlicek, J., Acton, S. T., Pattichis, M., 2001. Active contour segmentation guided by AM-FM dominant component analysis. In: Proceedings of the IEEE International Conference on Image Processing. Vol. 1. pp. 78–81.

Reed, T. R., du Buf, J. M. H., 1993. A review of recent texture segmentation and feature extraction techniques. CVGIP: Image Understanding 57 (3), 359–372.

Reyes-Aldasoro, C. C., Bhalerao, A., 2006. The Bhattacharyya space for feature selection and its application to texture segmentation. Pattern Recognition 39 (5), 812–826.

Rodríguez-Damián, M., Cernadas, E., Formella, A., Fernández-Delgado, M., De Sá-Otero, P., 2006. Automatic detection and classification of grains of pollen based on shape and texture. IEEE Transactions on Systems, Man, and Cybernetics–Part C 36 (4), 531–542.

Rohrmus, D. R., 2005. Invariant and adaptive geometrical texture features for defect detection and classification. Pattern Recognition 38 (10), 1546–1559.

Roselfeld, A., Lipkin, B. S., 1970. Texture synthesis. In: Lipkin, B. S., Rosenfeld, A. (Eds.), Picture Processing and Psychopictorics. Academic Press, pp. 309–322.

Rubner, Y., Tomasi, C., 1996. Coalescing texture descriptors. In: Proceedings of the ARPA Image Understanding Workshop. pp. 927–935.

Ruiz, R., Riquelme, J. C., Aguilar-Ruiz, J. S., 2006. Incremental wrapper-based gene selection from microarray data for cancer classification. Pattern Recognition 39 (12), 2383–2392.

Sabri, M., Alirezaie, J., 2004. Optimized space frequency kernel for texture classification. In: Proceedings of the IEEE International Conference on Image Processing. Vol. 3. pp. 1521–1524.

Saeys, Y., Inza, I., Larrañaga, P., 2007. A review of feature selection techniques in bioinformatics. Bioinformatics 23 (19), 2507–2517.

Sagiv, C., Sochen, N. A., Zeevi, Y. Y., 2000. Gabor-space geodesic active contours. In: Algebraic Frames for the Perception-Action Cycle, Lecture Notes in Computer Science. Vol. 1888/2000. pp. 309–318.

Sagiv, C., Sochen, N. A., Zeevi, Y. Y., 2006. Integrated active contours for texture segmentation. IEEE Transactions on Image Processing 15 (6), 1633–1646.

Sánchez, G., Lladós, J., 2001. A graph grammar to recognize textured symbols. In: Proceedings of the International Conference on Document Analysis and Recognition. pp. 465–469.

Sarkar, A., Biswas, M. K., Kartikeyan, B., Kumar, V., Majumder, K. L., Pal, D. K., 2002. A MRF model-based segmentation approach to classification for multispectral imagery. IEEE Transactions on Geoscience and Remote Sensing 40 (5), 1102–1113.

Sarkar, N., Chaudhuri, B. B., 1994. An efficient differential box-counting approach to compute fractal dimension of image. IEEE Transactions on Systems, Man, and Cybernetics 24 (1), 115–120.

Scarpa, G., Haindl, M., 2006. Unsupervised texture segmentation by spectral-spatial-independent clustering. In: Proceedings of the International Conference on Pattern Recognition. Vol. 2. pp. 151–154.

Schachter, B. J., Davis, L. S., Rosenfeld, A., 1979. Some experiments in image segmentation by clustering of local feature values. Pattern Recognition 11 (1), 19–28.

Schlimmer, J. C., 1993. Efficiently inducing determinations: a complete and systematic search algorithm that uses optimal pruning. In: Proceedings of the International Conference on Machine Learning. pp. 284–290.

Serrano, N., Savakis, A. E., Luo, J., 2004. Improved scene classification using efficient low-level features and semantic cues. Pattern Recognition 37 (9), 1773–1784.

Shafarenko, L., Petrou, M., Kittler, J., 1997. Automatic watershed segmentation of randomly textured color images. IEEE Transactions on Image Processing 6 (11), 1530–1544.

Shi, J., Malik, J., 2000. Normalized cuts and image segmentation. IEEE Transactions on Pattern Analysis and Machine Intelligence 22 (8), 888–905.

Simard, M., Saatchi, S. S., De Grandi, G., 2000. The use of decision tree and multiscale texture for classification of JERS-1 SAR data over tropical forest. IEEE Transactions on Geoscience and Remote Sensing 38 (5), 2310–2321.

Simoncelli, E. P., Adelson, E. H., 1990. Non-separable extensions of quadrature mirror filters to multiple dimensions. Proceedings of the IEEE 78 (4), 652–664.

Singh, M., Provan, G. M., 1996. Efficient learning of selective Bayesian network classifiers. In: Proceedings of the International Conference of Machine Learning. pp. 453–461.

Singh, S., Haddon, J., Markou, M., 2001. Nearest-neighbour classifiers in natural scene analysis. Pattern Recognition 34 (8), 1601–1612.

Sinha, A., Gupta, S., 2010. A fast nonparametric noncausal MRF-based texture synthesis scheme using a novel FKDE algorithm. IEEE Transactions on Image Processing 19 (3), 561–572.

Sivic, J., Zisserman, A., 2003. Video google: a text retrieval approach to object matching in videos. In: Proceedings of the IEEE International Conference on Computer Vision. pp. 1470–1477.

Smith, G., Burns, I., 1997. Measuring texture classification algorithms. Pattern Recognition Letters 18 (14), 1495–1501.

Sonka, M., Hlavac, V., Boyle, R., 1993. Analysis and Machine Vision. Chapman & Hall Computing.

Sumengen, B., Manjunath, B. S., 2007. Graph partitioning active contours (GPAC) for image segmentation. IEEE Transactions on Pattern Analysis and Machine Intelligence 28 (4), 509–521.

Super, B. J., Bovik, A. C., 1995. Shape from texture using local spectral moments. IEEE Transactions on Pattern Analysis and Machine Intelligence 17 (4), 333–343.

Szirányi, T., Csapodi, M., 1998. Texture classification and segmentation by cellular neural networks using genetic learning. Computer Vision and Image Understanding 71 (3), 255–270.

Tamura, H., Mori, S., Yamawaki, T., 1978. Textural features corresponding to visual perception. IEEE Transactions on Systems, Man, and Cybernetics SMC-8 (6), 460–473.

Tang, X., 1998. Texture information in run-length matrices. IEEE Transactions on Image Processing 7 (11), 1602–1609.

Teuner, A., Pichler, O., Hosticka, B. J., 1995. Unsupervised texture segmentation of images using tuned matched Gabor filters. IEEE Transactions on Image Processing 4 (6), 863–870.

Thompson, W. B., 1977. Textural boundary analysis. IEEE Transactions on Computers C-26 (3), 272–276.

Tolle, C. R., McJunkin, T. R., Gorsich, D. J., 2003. Suboptimal minimum cluster volume cover-based method for measuring fractal dimension. IEEE Transactions on Pattern Analysis and Machine Intelligence 25 (1), 32–41.

Torkkola, K., 2003. Feature extraction by non-parametric mutual information maximization. Journal of Machine Learning Research 3, 1415–1438.

Tsai, D.-M., Hsieh, C.-Y., 1999. Automated surface inspection for directional textures. Image and Vision Computing 18 (1), 49–62.

Tsai, D.-M., Tseng, C.-F., 1998. Surface roughness classification for castings. Pattern Recognition 32 (3), 389–405.

Tsai, D.-M., Wu, S.-K., Chen, M.-C., 2001. Optimal Gabor filter design for texture segmentation using stochastic optimization. Image and Vision Computing 19 (5), 299–316.

Tsuji, S., Tomita, F., 1973. A structural analyzer for a class of textures. Computer Graphics and Image Processing 2, 216–231.

Tsujinishi, D., Koshiba, Y., Abe, S., 2004. Why pairwise is better than one-against-all or all-at-once. In: Proceedings of the International Joint Conference on Neural Networks. pp. 693–698.

Tuceryan, M., Jain, A. K., 1990. Texture segmentation using voronoi polygons. IEEE Transactions on Pattern Analysis and Machine Intelligence 12 (2), 211–216.

Tuceryan, M., Jain, A. K., 1998. Texture analysis. In: Chen, C. H., Pau, L. F., Wang, P. S. P. (Eds.), Handbook of Pattern Recognition and Computer Vision. World Scientific Publishing Company, pp. 207–248.

Turner, M. R., 1986. Texture discrimination by Gabor functions. Biological Cybernetics 55 (11), 71–82.

Unal, G., Yezzi, A., Krim, H., 2005. Information-theoretic active polygons for unsupervised texture segmentation. International Journal of Computer Vision 62 (3), 199–220.

Unser, M., 1995. Texture classification and segmentation using wavelet frames. IEEE Transactions on Image Processing 4 (11), 1549–1560.

Unser, M., de Coulon, F., 1982. Detection of defects by texture monitoring in automatic visual inspection. In: Proceedings of the International Conference on Robot Vision and Sensory Controls. pp. 27–38.

Vafaie, H., Bourbakis, N. G., 1988. A tree grammar scheme for generation and recognition of simple texture paths in pictures. In: Proceedings of the IEEE International Symposium on Intelligent Control. pp. 201–206.

Vafaie, H., Imam, I. F., 1994. Feature selection methods: genetic algorithms vs. greedy-like search. In: Proceedings of the International Conference on Fuzzy and Intelligent Control Systems.

Vanrell, M., 1996. Identificació de les Dimensions d'un Espai de Representació de Textures Basat en un Model Computacional de Percepció Preatentiva. Ph.D. thesis, Departament d'Informàtica, Universitat Autònoma de Barcelona.

Vapnik, V., 1998. Statistical Learning Theory. John Wiley & Sons.

Varma, M., Zisserman, A., 2005. A statistical approach to texture classification from single images. International Journal of Computer Vision 62 (1/2), 61–81.

Wang, K., Bell, D., Murtagh, F., 1998. Relevance approach to feature subset selection. In: Liu, H., Motoda, H. (Eds.), Feature Extraction, Construction and Selection. Kluwer Academic Publishers, pp. 85–97.

Weldon, T. P., 2006. Improved image segmentation with a modified Bayesian classifier. In: Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing. Vol. 2. pp. 697–700.

Weldon, T. P., Higgins, W. E., 1996. Design of multiple Gabor filters for texture segmentation. In: Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing. Vol. 4. pp. 2243–2246.

Weldon, T. P., Higgins, W. E., Dunn, D. F., 1996. Efficient Gabor filter design for texture segmentation. Pattern Recognition 29 (12), 2005–2015.

Weston, J., Elisseeff, A., Schölkopf, B., Tipping, M., 2003. Use of the zero-norm with linear models and kernel methods. Journal of Machine Learning Research 3, 1439–1461.

Weszka, J. S., Dyer, C. R., Rosenfeld, A., 1976. A comparative study of texture measures for terrain classification. IEEE Transactions on Systems, Man, and Cybernetics SMC-6, 269–285.

Weszka, J. S., Rosenfeld, A., 1978. Threshold evaluation techniques. IEEE Transactions on Systems, Man, and Cybernetics SMC-8 (8), 622–629.

Wilson, R., Calway, A. D., Pearson, E. R. S., 1992. A generalized wavelet transform for Fourier analysis: the multiresolution Fourier transform and its application to image and audio signal analysis. IEEE Transactions on Information Theory 38 (2), 674–690.

Winn, J., Criminisi, A., Minka, T., 2005. Object categorization by learned universal visual dictionary. In: Proceedings of the IEEE International Conference on Computer Vision. pp. 1800–1807.

Won, C. S., Derin, H., 1992. Unsupervised segmentation of noisy and textured images using Markov random fields. CVGIP: Graphical Models and Image Processing 54 (4), 308–328.

Wornell, G. W., Oppenheim, A. V., 1992. Estimation of fractal signals from noisy measurements using wavelets. IEEE Transactions on Signal Processing 40 (3), 611–623.

Wu, C.-C., Wang, Z.-F., 2006. Stereo correspondence using stripe adjacency graph. In: Proceedings of the International Conference on Pattern Recognition. Vol. 1. pp. 123–126.

Wu, C.-M., Chen, Y.-C., 1992. Statistical feature matrix for texture analysis. CVGIP: Graphical Models and Image Processing 54 (5), 407–419.

Wu, W.-R., Wei, S.-C., 1996. Rotation and gray-scale transform-invariant texture classification using spiral resampling, subband decomposition, and hidden Markov model. IEEE Transactions on Image Processing 5 (10), 1423–1434.

Wu, Z., Leahy, R., 1993. An optimal graph theoretic approach to data clustering: theory and its application to image segmentation. IEEE Transactions on Pattern Analysis and Machine Intelligence 11 (9), 1101–1113.

Xia, Y., Feng, D. D., Zhao, R., 2006a. Morphology-based multifractal estimation for texture segmentation. IEEE Transactions on Image Processing 15 (3), 614–623.

Xia, Y., Feng, D. D., Zhao, R., 2006b. Adaptive segmentation of textured images by using the coupled Markov random field model. IEEE Transactions on Image Processing 15 (11), 3559–3566.

Xu, L., Yan, P., Chang, T., 1988. Best first strategy for feature selection. In: Proceedings of the International Conference on Pattern Recognition. pp. 706–708.

Yang, A., Wright, J., Ma, Y., Sastry, S., 2008. Unsupervised segmentation of natural images via lossy data compression. Computer Vision and Image Understanding 110 (2), 212–225.

Yang, L., Meer, P., Foran, D. J., 2007. Multiple class segmentation using a unified framework over mean-shift patches. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 1–8.

You, J., Cohen, H. A., 1993. Classification and segmentation of rotated and scaled textured images using texture "tuned" masks. Pattern Recognition 26 (2), 245–258.

Yu, L., Liu, H., 2004. Efficient feature selection via analysis of relevance and redundancy. Journal of Machine Learning Research 5, 1205–1224.

Zhang, H., Berg, A. C., Maire, M., Malik, J., 2006. SVM-KNN: discriminative nearest neighbor classification for visual category recognition. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. Vol. 2. pp. 2126–2136.

Zhang, J., Tan, T., 2002. Brief review of invariant texture analysis methods. Pattern Recognition 35 (3), 735–747.

Zhu, S. C., Yuille, A., 1996. Region competition: unifying snakes, region growing and Bayes/MDL for image segmentation. IEEE Transactions on Pattern Analysis and Machine Intelligence 18 (9), 884–900.

Zöller, T., Buhmann, J. M., 2007. Robust image segmentation using resampling and shape constraints. IEEE Transactions on Pattern Analysis and Machine Intelligence 29 (7), 1147–1164.

Zucker, S. W., 1976. Toward a model of texture. Pattern Recognition 5 (2), 190–202.