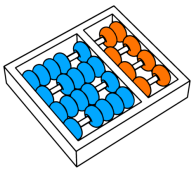
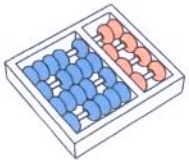


Fernando Roberti de Siqueira

"Multi-Scale Approaches to Texture Description"

"*Abordagens Multiescala para Descrição de Textura*"

CAMPINAS
2013



University of Campinas
Institute of Computing

Universidade Estadual de Campinas
Instituto de Computação

Fernando Roberti de Siqueira

"Multi-Scale Approaches to Texture Description"

Supervisor: **Prof. Dr. Hélio Pedrini**
Orientador(a):

Co-Supervisor: **Prof. Dr. William Robson Schwartz**
Co-orientador(a):

"Abordagens Multiescala para Descrição de Textura"

MSc Dissertation presented to the Post Graduate Program of the Institute of Computing of the University of Campinas to obtain a Master degree in Computer Science.

Dissertação de Mestrado apresentada ao Programa de Pós-Graduação em Ciência da Computação do Instituto de Computação da Universidade Estadual de Campinas para obtenção do título de Mestre em Ciência da Computação.

THIS VOLUME CORRESPONDS TO THE FINAL VERSION OF THE DISSERTATION DEFENDED BY FERNANDO ROBERTI DE SIQUEIRA, UNDER THE SUPERVISION OF PROF. DR. HÉLIO PEDRINI.

ESTE EXEMPLAR CORRESPONDE À VERSÃO FINAL DA DISSERTAÇÃO DEFENDIDA POR FERNANDO ROBERTI DE SIQUEIRA, SOB ORIENTAÇÃO DE PROF. DR. HÉLIO PEDRINI.

Hélio Pedrini

Supervisor's signature / *Assinatura do Orientador(a)*

CAMPINAS
2013

Ficha catalográfica
Universidade Estadual de Campinas
Biblioteca do Instituto de Matemática, Estatística e Computação Científica
Maria Fabiana Bezerra Muller - CRB 8/6162

Si75m

Siqueira, Fernando Roberti de, 1989-
Multi-scale approaches to texture description / Fernando Roberti de Siqueira. – Campinas, SP : [s.n.], 2013.

Orientador: Hélio Pedrini.
Coorientador: William Robson Schwartz.
Dissertação (mestrado) – Universidade Estadual de Campinas, Instituto de Computação.

1. Abordagem multiescala. 2. Descritores. 3. Processamento de imagens. I. Pedrini, Hélio,1963-. II. Schwartz, William Robson. III. Universidade Estadual de Campinas. Instituto de Computação. IV. Título.

Informações para Biblioteca Digital

Título em outro idioma: Abordagens multiescala para descrição de textura
Palavras-chave em inglês:
Multiscale approach
Descriptors
Image processing
Área de concentração: Ciência da Computação
Titulação: Mestre em Ciência da Computação
Banca examinadora:
Hélio Pedrini [Orientador]
Moacir Pereira Ponti Junior
Anderson de Rezende Rocha
Data de defesa: 15-10-2013
Programa de Pós-Graduação: Ciência da Computação

TERMO DE APROVAÇÃO

Dissertação Defendida e Aprovada em 15 de outubro de 2013, pela Banca examinadora composta pelos Professores Doutores:

Prof. Dr. Moacir Pereira Ponti Junior
ICMC / USP

Prof. Dr. Anderson de Rezende Rocha
IC / UNICAMP

Prof. Dr. Hélio Pedrini
IC / UNICAMP

Institute of Computing /*Instituto de Computação*
University of Campinas /*Universidade Estadual de Campinas*

Multi-Scale Approaches to Texture Description

Fernando Roberti de Siqueira

October 15, 2013

Examiner Board/*Banca Examinadora*:

- Prof. Dr. Hélio Pedrini (Supervisor/*Orientador*)
- Prof. Dr. Moacir Pereira Ponti Junior
  ICMC - USP
- Prof. Dr. Anderson de Rezende Rocha
  IC - UNICAMP
- Prof. Dr. Marcelo da Silva Hounsell
  DCC - UDESC (Substitute/*Suplente*)
- Prof. Dr. Ricardo da Silva Torres
  IC - UNICAMP (Substitute/*Suplente*)

Abstract

Computer vision and image processing techniques play an important role in several fields, including object detection and image classification, which are very important tasks with applications in medical imagery, remote sensing, forensic analysis, skin detection, among others. These tasks strongly depend on visual information extracted from images that can be used to describe them efficiently. Texture is one of the main used characteristics that describes information such as spatial distribution, brightness and surface structural arrangements. For image recognition and classification, a large set of texture descriptors was investigated in this work, such that only a small fraction is actually multi-scale. Gray level co-occurrence matrices (GLCM) have been widely used in the literature and are known to be an effective texture descriptor. However, such descriptor only discriminates information on a unique scale, that is, the original image. Scales can offer important information in image analysis, since texture can be perceived as different patterns at distinct scales. For that matter, two different strategies for extending the GLCM to multiple scales are presented: (i) a Gaussian scale-space representation, constructed by smoothing the image with a low-pass filter and (ii) an image pyramid, which is defined by sampling the image both in space and scale. This texture descriptor is evaluated against others in different data sets. Then, the proposed texture descriptor is applied in skin detection context, as a mean of improving the accuracy of the detection process. Experimental results demonstrated that the GLCM multi-scale extension has remarkable improvements on tested data sets, outperforming many other feature descriptors, including the original GLCM.

Resumo

Visão computacional e processamento de imagens desempenham um papel importante em diversas áreas, incluindo detecção de objetos e classificação de imagens, tarefas muito importantes para aplicações em imagens médicas, sensoriamento remoto, análise forense, detecção de pele, entre outras. Estas tarefas dependem fortemente de informação visual extraída de imagens que possa ser utilizada para descrevê-las eficientemente. Textura é uma das principais propriedades usadas para descrever informação tal como distribuição espacial, brilho e arranjos estruturais de superfícies. Para reconhecimento e classificação de imagens, um grande grupo de descritores de textura foi investigado neste trabalho, sendo que apenas parte deles é realmente multiescala. Matrizes de coocorrência em níveis de cinza (GLCM) são amplamente utilizadas na literatura e bem conhecidas como um descritor de textura efetivo. No entanto, este descritor apenas discrimina informação em uma única escala, isto é, a imagem original. Escalas podem oferecer informações importantes em análise de imagens, pois textura pode ser percebida por meio de diferentes padrões em diferentes escalas. Dessa forma, duas estratégias diferentes para estender a matriz de coocorrência para múltiplas escalas são apresentadas: (i) uma representação de escala-espaço Gaussiana, construída pela suavização da imagem por um filtro passa-baixa e (ii) uma pirâmide de imagens, que é definida pelo amostragem de imagens em espaço e escala. Este descritor de textura é comparado com outros descritores em diferentes bases de dados. O descritor de textura proposto é então aplicado em um contexto de detecção de pele, como forma de melhorar a acurácia do processo de detecção. Resultados experimentais demonstram que a extensão multiescala da matriz de coocorrência exibe melhora considerável nas bases de dados testadas, exibindo resultados superiores em relação a diversos outros descritores, incluindo a versão original da matriz de coocorrência em escala única.

Contents

Chapter 1

Introduction

With the increasing usage of the Internet as an everyday tool for billions of people, approximately 300 million images were uploaded every day in 2012 [92] and 100 hours of video were added per minute on the main social media [129] in 2013. Those numbers demonstrate that users are providing an enormity of visual content. The problem is that, among these uploads, a lot of malicious information can become available as well, which can lead to serious issues, such as child pornography, violation of copyright, pornography open to general public, among others.

How can this content be filtered in a way that it does not reach the general public? Many websites perform various kinds of validation on the uploaded content. Manual verification is prohibitive due to the large volume of information, once no single team can be responsible for labeling the content as appropriate or inappropriate. One possibility, that is applied in several sites, is a button where the users can flag that content as inappropriate. Although being quite helpful, a better solution should be used, that would not expose the users to troublesome data, even if just a small percentage of them. The need for an automatic procedure to filter the uploaded data with a high degree of accuracy is obvious, which can be done through image classification and analysis techniques.

Furthermore, a user expecting to find a specific picture or related images will have enormous difficulty in searching the content. A fast and easy way to browse through data is required, pursuing methods for storing and easily retrieving information, providing better usability for the final user. Such investigation field is named Content-Based Image Retrieval (CBIR) [4, 56, 87, 96] an automatic process for searching relevant images to a given query usually based on low-level features such as color, texture, shape and spatial layout. The basis of the involved image processing techniques is the usage of an entity that represents the images, where the main purpose is to avoid the entire image to be used for all operations, such that only the most relevant characteristics extracted by this simple entity are employed. This makes the storage and retrieval processes more efficient.

Both image classification and CBIR can be addressed through feature descriptors, since they provide an efficient way to describe images not demanding much space and avoiding the use of an entire image in processing operations. A feature descriptor is a mathematical representation of an object or image region under analysis, in such a way that most of the necessary information for proper distinction is preserved, while maintaining a smaller number of characteristics than the object itself possesses. Its purpose is to describe the content of an entire image efficiently, so that images with similar characteristics or properties will have similar descriptors.

A simple approach to designing a descriptor is to employ a structure to detect patterns often repeated by certain objects. For example, the image of a wooden surface contains variations of intensities which form a repeated pattern, known as visual texture [116]. The texture of an object comes from an abstract interpretation, hence it is information acquired by human senses in a very coarse way and, therefore, lacks a proper formal definition. Textures appear all around our lives, everything that we can perceive through our eyes is filled with different kinds of textures that possess various properties of their own [115]. The perception of texture is believed to play an important role in the human visual system for recognition and interpretation. Trying to take advantage of what our eyes naturally evolved to do, texture image processing focuses on visual patterns repeated throughout a region, such as orientation, structural arrangements, changes in intensity or color brightness.

A wide variety of texture measures have been proposed in the literature, ranging from very distinct approaches such as statistical [34, 40, 41], parametric models [22, 68], signal processing [26, 27, 66], geometric [42, 43, 57, 75], among others. Several comparative studies to evaluate the performance of some measures have been carried out by a number of authors in the literature, such as Weszka et al. [124], Du Buf et al. [30], Ohanian and Dubes [73], Timo Ojala et al. [76], among many others. Researches have attempted to address other difficulties inherent to texture information, such as different orientation and scale, changes in illumination and resolution, as well as different types of noises or other imperfections. Various techniques incorporating invariance (scale-wise or rotation-wise) have attempted to neutralize such issues [2, 63, 67].

The main purpose of this work is to investigate the use of texture descriptors in image analysis and to propose a method for improving their description capability of perceiving image information through multiple scales. The results achieved by enhancing the texture descriptor help to improve its use in many image processing techniques, such as classification and CBIR along with their applications. As a consequence, in improving existing features or in creating a new accurate feature, all related fields might be improved as well, hence image description is the foundation for most of the operations performed in image processing and computer vision areas.

Finally, the proposed multi-scale texture descriptor was applied in human skin detection to evaluate its gain in such context. Skin detection can be incorporated as a stage in an analysis process to identify, for instance, pornography in images or videos or be used to improve people tracking, among other applications. Several skin detection methods have been described in the literature based on color threshold values, such as [20, 35, 54], some based on histogram classification methods, called normalized lookup table [53], Gaussian modeling [1, 46], and many others [61, 89]. To further explore this scenario, an experiment was performed by using a normalized lookup table in an adaptable configuration, as proposed in [90], and a step of texture detection was added to improve the results of the lookup table detection. LBP, GLCM and the multi-scale GLCM extension were evaluated in this texture detection step.

The three main topics investigated in this dissertation are: feature descriptors for texture classification [103], multi-scale texture descriptors [108], and skin detection based on texture information [109].

The text is organized as follows. Chapter 2 presents an overview with concepts and work related to this dissertation. Chapter 3 presents a survey of many texture descriptors of the literature and an evaluation of their results in several public texture data sets. Chapter 4 describes a method for extending a widely used single-scaled texture descriptor to a multi-scale strategy, as well as its respective result enhancements assessed in several data sets. The main concepts and work related to skin detection problem are exposed in Chapter 5, where an adaptive approach is employed to detect human skin and compared to other texture descriptors. Finally, Chapter 6 concludes this work with final remarks and directions for future work.

Chapter 2

Concepts and Related Work

This chapter describes concepts and applications of texture analysis, how they can be used in skin detection and our approach to creating a multi-scale descriptor.

2.1 Texture Analysis

Texture is a very difficult concept to describe precisely, once it is related to subjective sensory perception such as tactile, gustation and vision. As tactorial, the information is usually described in terms of smoothness and roughness. Even the visual information seems to be an expression of how the visual aspect of the object would be if touched. Other visual texture elements exist, such as shininess and opaqueness. In image processing field, these perceptual aspects are used to express texture in images with properties related to structural arrangement of surfaces or changes in intensity or color brightness.

Human beings have developed throughout the years mechanisms to differentiate several types of texture. Image analysis field has strongly attempted to translate such mechanisms into a large number of descriptors used in the literature [40, 59, 99, 112, 113, 127].

A feature descriptor extracts visual information from images in a concise way, retaining as much information as possible. It is usually represented by vectors, denominated as feature vectors. Essentially, this descriptor can be a set of statistical information, parametric models, coefficients obtained after image transforms or even combination of such measures.

In this work, the taxonomy used to categorize the texture analysis techniques follows the work by Tuceryan and Jain [116]: geometric, parametric, signal processing and statistical models.

One important usage of texture is in image classification, which is usually divided into two steps, a learning stage and a recognition stage. In the first step, a model is built to describe the texture content of each class for the available training data, whereas

in the second step, information from an unknown source is extracted and compared to those previously trained. This sample is then labeled according to the best match among those obtained in the first stage. A comparison among a large set of texture features, by assessing which one is more suitable for texture classification, is presented in the next chapter, such that classification rates are compared for numerous data sets.

2.2 Multiple Scales of Information

The multiscale or multi-resolution analysis allows to study a signal or function at different scales. Scales are important in image processing since the extraction of small elements from the images could be restricted to details, while at coarser scales other information such as contour or determining shapes become predominant. Therefore, the multi-scale analysis of a signal is recommended for interpreting it more effectively [6].

From a combination of features at each of these scales, a more efficient descriptor can be extracted from images, mixing the characteristics that are only perceptible at coarser scales to details at finer scales.

The Gaussian pyramid representation [86] is formed by low-pass filtered versions of a Gaussian convoluted image. The Laplacian pyramid [17, 18, 23], on the other hand, is formed by the decomposition of the image by means of high-pass filters. The lowest level of the pyramid contains only the higher frequency components (for example, image edges), while subsequent levels contain components to lower frequencies.

The wavelet transforms [25,26,66] decompose signals allowing recovery of both spatial information and frequency information. When the transform is applied to two-dimensional images, the wavelet functions are applied as high-pass filters, while the scaling functions act as low-pass filters. This process is performed recursively for the sub-image that has a lower frequency, such that the original image is itself decomposed into a series of images with different scales.

The Gabor transform [27,98] also allows the characterization of a signal in spatial and frequency domains. The transform analyzes the signal in a Gaussian window, such that the resolution in space and frequency remains approximately constant for the window. Although Gabor transform presents higher computational cost compared to many other wavelet transforms, it allows for a relatively simple interpretation of the results and flexibility in controlling the orientation information and scale, enabling the development of descriptors invariant to changes in scale and rotation.

In many descriptors, it is assumed that texture information is fixed at a specific image resolution. The discriminative power of texture descriptors can be significantly improved if different scales are considered among the images during the descriptor extraction.

Among the statistical approaches, gray level co-occurrence matrices (GLCM) [5, 7, 41,

69, 91] demonstrated to be a useful texture descriptor used in image analysis. However, the original GLCM has limited capability of capturing texture information at multiple scales, which is an opportunity for improvement.

This work presents a novel scheme for extending the GLCM to be more robust by using information of various different scales. Two different multi-scale representations are used in the extension of the descriptor. The performance of the proposed approach is evaluated by applying the multi-scale descriptor on five benchmark texture data sets and the results are compared to other powerful texture operators.

2.3 Skin Detection

Human skin detection techniques have important applications in many areas, such as gesture analysis, nudity detection, content-based image retrieval, face identification. In a general way, the presence of people in an image or a video scene can be evidenced by finding skin regions.

The proposition of an automatic skin detection process still presents several challenges, especially under varying illumination and partial occlusions [88]. Another inherent difficulty is that skin tones can significantly vary across individuals.

Color information has been extensively explored in skin detection methods. Several color spaces or models have been employed and compared in order to determine their importance in the detection process e their capability to overcome the associate challenges. Elgammal et. al. [31] developed a comparative study among different color spaces and their ability to represent distinct skin tones. Zarit et. al. [130] compared five color spaces for skin modeling. Shin et. al. [105] investigated the effect of color space transformations on increasing separability between skin and non-skin classes, where results indicate that most of the transformations do not present the assumed benefits.

An analysis of works available in the literature shows that there are variants on how skin can be detected in an image [61, 89, 120]. Fixed thresholds can be used with almost any skin tone and detect most of the skin portions in the image. Additionally, relaxing the thresholds can lead to more robust ways to identify skin under image resolution changes and geometric variations of skin patterns [20, 35, 54]. Normalized lookup tables are based on histograms acquired on sets of training images, providing probability of a particular color tone to be skin [53]. Gaussian Model is a skin color distribution model that can define how "skin-like" the tested color is based on an elliptic Gaussian joint probability density function [70]. Alternatively, the Mahalanobis distance from the color vector to the mean vector can be used [1, 46].

An enhancement of the adaptive normalized lookup table [90], whose resulting probability map is used to detect skin and non-skin regions in the images, is proposed by adding

a texture detection step, used to refine its results. Experiments were conducted to apply the methodology to several images and results show the effectiveness of the proposed method.

Chapter 3

Evaluation of Feature Descriptors for Texture Classification

Preamble

Successful execution of tasks such as image classification, object detection and recognition, and scene classification depends on the definition of a set of features able to describe images effectively. Texture is among the features used by the human visual system. It provides information regarding spatial distribution, changes in brightness, and description regarding the structural arrangement of surfaces. However, although the visual human system is extremely accurate to recognize and describe textures, it is difficult to define a set of textural descriptors to be used in image analysis on different application domains. This work evaluates several texture descriptors and demonstrates that the combination of descriptors can improve the performance of texture classification.

3.1 Introduction

The definition of a set of visual features able to describe images effectively, so that classification, detection and recognition processes can be applied, is a complex task in image analysis. A way to address this problem is to recur to features used by humans to understand visual information.

Texture is among the features used by the human visual system and can be characterized by local variations of pixel values that repeat in a regular or random pattern on the object or image. It can also be defined as a repetitive arrangement of patterns over a region [102]. It provides information regarding spatial distribution, changes in brightness,

and description regarding the structural arrangement of surfaces. Therefore, the use of textural features is an important source of information for image description.

Although the visual human system is extremely accurate to recognize and describe textures, it is difficult to define a set of textural descriptors to be used in image analysis on different application domains, or even to formalize a definition for texture. Such difficulty is reflected by the large number of definitions and descriptors found in the literature [40, 59, 99, 112, 113, 127].

Feature descriptors are extracted from the input image and can be based on second-order statistics, parametric models, coefficients obtained from an image transform, or even a combination of these measures.

Texture classification usually involves two main stages, the learning step and the recognition step. In the first stage, a model is built to represent the texture content of each class present in the training data. In the second stage, the texture content of an unknown sample is extracted and compared to those extracted in the learning step. The sample is labeled to the class with the best match.

This work describes and compares a large set of feature descriptors in order to assess which are more suitable to be applied to texture classification. Furthermore, an experimental evaluation is presented to demonstrate that the combination of features produces superior results in terms of classification rate when compared to the individual use of the features, which means that some of them are complementary.

The paper is organized as follows. Section 5.2 presents texture descriptors considered in the evaluation. Experimental results are shown and discussed in Section 5.4. Finally, Section 6 concludes the paper with final remarks.

3.2 Texture Descriptors

This section describes a number of relevant methods for texture feature extraction. Even though there is no a unique taxonomy to classify such methods, this work categorizes them into the following categories: *statistical approach* (Section 3.2.1), *approach based on signal processing* (Section 3.2.2), *geometrical approach* (Section 3.2.3), and *approach based on parametric models* (Section 3.2.4). This taxonomy is based on that proposed by Tuceryan and Jain [116].

3.2.1 Statistical Approach

Methods based on the statistical approach do not explore hierarchical structures presented by the texture, but represent its properties in indirect and probabilistic manners.

The simplest primitive that can be defined in a grayscale digital image is the pixel, which has the gray level as property. Consequently, the gray level distribution could be described by first order statistics, such as mean and variance estimated from a histogram computed from this distribution. However, since the first order statistics consider only pixels individually, this makes such measures more sensitive to changes in the image. Therefore, to avoid this problem, second order statistics, which depend on transitions between gray level of pixels, are considered.

The following sections describe the main statistical methods for texture analysis, including those based on first order statistics, co-occurrence matrix, features extracted from higher order statistics, such as gray level run length matrices, and autocorrelation function.

First Order Statistics

From the gray level histogram of a textured image, it is possible to extract first order statistics. Given an image with n pixels, the histogram can be computed using Equation 3.1, where $h(i)$ represents the number of occurrences of the i-th gray level.

$$P(i) = \frac{h(i)}{n} \tag{3.1}$$

Although first order statistics present disadvantages, the computational cost to extract descriptors is very low since only simple measures, such as mean, variance, skewness, and kurtosis, need to be computed. These measures are shown in Equations 3.2 to 3.5, respectively, where H_g denotes the largest gray level in the image.

$$\mu = \frac{1}{n}\sum_{i=0}^{H_g} h(i) \tag{3.2}$$

$$\sigma^2 = \frac{1}{n-1}\sum_{i=0}^{H_g} (h(i)-\mu)^2 \tag{3.3}$$

$$s = \frac{1}{n\sigma^3}\sum_{i=0}^{H_g} (h(i)-\mu)^3 \tag{3.4}$$

$$k = \frac{1}{n\sigma^4}\sum_{i=0}^{H_g} (h(i)-\mu)^4 - 3 \tag{3.5}$$

Energy (Equation 3.6) and entropy (Equation 3.7) are also other measures computed

from the image histogram.

$$E = \sum_{i=0}^{H_g} (P(i))^2 \tag{3.6}$$

$$H = -\sum_{i=0}^{H_g} P(i) \log(P(i)) \tag{3.7}$$

Gray Level Co-occurrence Matrix

An approach to extracting textural information regarding gray level transition between two pixels uses a co-occurrence matrix. Given a spatial relationship defined among pixels in a texture, such matrix represents the joint distribution of gray-level pairs of neighboring pixels. Therefore, matrices providing different information are obtained by modifying the spatial relationship (different orientation or distance between pixels). Descriptors are extracted from these matrices.

The number of rows and columns of the co-occurrence matrix depends only on the gray levels in the texture and not on the image size. The element $P(m, n)$ of a co-occurrence matrix indicates the number of transitions between the gray level m and n that take place in the texture according to a given spatial relationship.

Before computing the co-occurrence matrix, it is necessary to define relations among pixels, that is, the arrangement of pixels from which the transitions will be considered. A set S is built. Each element in this set is a pair of coordinates of each pixel involved in the relationship. Once S is defined, Equation 3.8 is used to count the number of transitions between each pair of gray levels in the texture. In this equation, $f(x, y)$ indicates the gray level of a pixel located at (x, y) in the image.

$$P(m, n) = \#\{((i, j), (k, l)) \in S \mid f(i, j) = m \text{ and } f(k, l) = n\} \tag{3.8}$$

Once the frequency of each gray level transition is computed, $P(m, n)$ is placed at the m-th row and n-th column of the matrix. Then, feature descriptors are extracted after a normalization based on Equation 3.9, where H_g denotes the largest gray level.

$$p_{m,n} = \frac{P(m, n)}{\sum_{i=0}^{H_g} \sum_{j=0}^{H_g} P(i, j)} \qquad m, n = 0, \ldots, H_g \tag{3.9}$$

According to Equation 3.8, the co-occurrence matrix depends on the gray level transitions between pairs of pixels in set S. This way, it is possible to arbitrarily specify the distance and the angle between the pairs. Haralick et al. [41] defined specifically which

transitions should be considered to compute co-occurrence matrices. Two additional parameters are included, d and θ. These parameters define the distance and angle between pixels in S, respectively. Therefore, several matrices may be obtained with small changes in these parameters. To describe the properties contained in the co-occurrence matrices, Haralick et al. proposed 14 statistical measures that are computed from the matrices: angular second moment, contrast, correlation, sum of squares, inverse difference moment, sum average, sum variance, sum entropy, entropy, difference variance, difference entropy, two information measures of correlation, and maximal correlation coefficient.

Gray Level Run Length Matrices

Gray level runs are obtained by sampling co-linear regions with the same gray level in an image. Aiming at summarizing the information obtained by the runs, Galloway [34] proposed a matrix to tabulate the number of runs with specific lengths for given gray levels. Consequently, high order statistics computed from this matrix can be used for texture analysis.

From these matrices, called gray level run length matrices (GLRLM), relevant information regarding the texture can be extracted. It is expected long runs to be more frequent in coarse textures, while short runs are expected in fine textures due to edges and abrupt changes in the gray level.

The matrices proposed by Galloway are built as follows. Given a fixed orientation θ, the set composed by consecutive pixels with the same gray level on this orientation represents a gray level run and the number of pixels in this run is denoted by run length. Then, each entry of the GLRLM, $P(i, j|\theta)$, contains the number of runs with length j with gray level i for an orientation θ. Although the orientation θ can assume any value, in general the GLRLM are computed for a subset of orientations: $0^o, 45^o, 90^o$ and 135^o. Figure 3.1 illustrates the runs for these orientations.

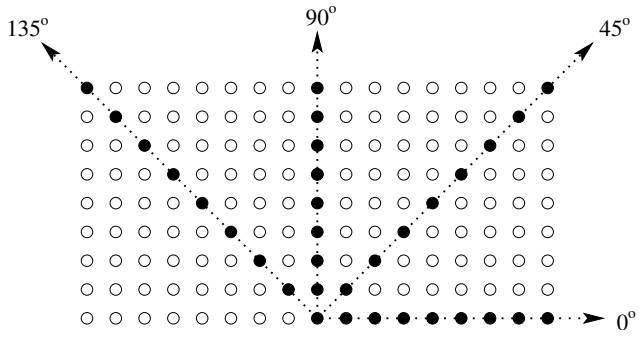


Figure 3.1: Orientations from which the GLRLM are usually computed.

Once the GLRLM are available, measures are computed and used as descriptors. The

measures proposed by Galloway are described as follows. In these equations, H_g and N_r denote the largest gray level and the largest run length, respectively.

Short runs emphasis (SRE) and long runs emphasis (LRE), shown in Equations 3.10 and 3.11. LRE presents large values when large run lengths are present in the texture.

$$\mathrm{SRE} = \sum_{i=0}^{H_g}\sum_{j=1}^{N_r} \frac{P(i,j|\theta)}{j^2} \Big/ \sum_{i=0}^{H_g}\sum_{j=1}^{N_r} P(i,j|\theta) \tag{3.10}$$

$$\mathrm{LRE} = \sum_{i=0}^{H_g}\sum_{j=1}^{N_r} j^2 P(i,j|\theta) \Big/ \sum_{i=0}^{H_g}\sum_{j=1}^{N_r} P(i,j|\theta) \tag{3.11}$$

Measures of gray level non-uniformity (GLN) and run length non-uniformity (RLN), shown in Equations 3.12 and 3.13, respectively, are used to describe the distribution of gray levels and run lengths. For example, GLN presents small values when the number of runs is uniformly distributed according to the gray level.

$$\mathrm{GLN} = \sum_{i=0}^{H_g} \left(\sum_{j=1}^{N_r} P(i,j|\theta) \right)^2 \Big/ \sum_{i=0}^{H_g}\sum_{j=1}^{N_r} P(i,j|\theta) \tag{3.12}$$

$$\mathrm{RLN} = \sum_{i=1}^{N_r} \left(\sum_{j=0}^{H_g} P(i,j|\theta) \right)^2 \Big/ \sum_{i=0}^{H_g}\sum_{j=1}^{N_r} P(i,j|\theta) \tag{3.13}$$

The last measure proposed by Galloway, called run percentage (RP), is shown in Equation 3.14, where n denotes the number of pixels in the image. This measure presents large values when the texture is mostly composed of short runs.

$$\mathrm{RP} = \frac{1}{n} \sum_{i=0}^{H_g}\sum_{j=1}^{N_r} P(i,j|\theta) \tag{3.14}$$

Autocorrelation Function

An approach to discriminating between coarse and fine textures is based on spatial frequency. Fine textures are composed of small primitives and therefore present high spatial frequency due to the large number of gray level variations, whereas coarse textures possess large primitives with low spatial frequency.

The autocorrelation function describes spatial interactions between the primitives composing a texture [40]. In this case, the gray levels are assumed to be the primitives that compose a texture, while the interactions among these primitives are characterized by the autocorrelation coefficient. This coefficient is obtained by Equation 3.15 for a texture with $M \times N$ pixels. For each pair of values $\{p, q\}$, the texture is shifted by at most p

pixels in x coordinate and q pixels in y coordinate.

$$\rho_{ff}(p,q) = \frac{\sum_{i=0}^{M-p-1}\sum_{j=0}^{N-q-1} f(i,j)f(i+p,j+q)}{\sum_{i=0}^{M-1}\sum_{j=0}^{N-1} f^2(i,j)} \tag{3.15}$$

3.2.2 Approach Based on Signal Processing

Texture analysis methods based on signal processing extract descriptors from an image representation obtained by applying image transforms, such as Fourier or wavelet transforms.

Fourier Spectrum

The spectrum resulting from the 2D Fourier transform (shown in Equation 3.16 for a $n \times n$ image, where $i = \sqrt{-1}$), after shifting the frequency plane from the origin, shows high energy concentration at the center when the image presents low spatial frequency, whereas the energy is more spread when the image has high frequency. Extending this concept to texture analysis, coarse textures will present high concentration of energy at the center due to homogeneity. On the other hand, the energy will be spread around the plane when fine textures are considered.

$$\mathcal{F}(u,v) = \frac{1}{n^2}\sum_{k=0}^{n-1}\sum_{l=0}^{n-1} f(k,l)\exp(-2\pi i(ku+lv)/n) \tag{3.16}$$

Converting the Fourier spectrum, S, to a polar representation $S(r,\theta)$, r and θ are the variables in this coordinate system. For each direction θ, $S(r,\theta)$ can be considered as a function $S_\theta(r)$ and, similarly, for each radius r, $S_r(\theta)$ is also a unidimensional function. Therefore, the analysis of $S_\theta(r)$ for a given value of θ gives the behavior of the spectrum along with a radial direction and the analysis of $S_r(\theta)$ for a given value of r explains the behavior of the spectrum along a circle centered at the origin.

A global description is given by functions shown in Equations 3.17 and 3.18. Unidimensional descriptions of the spectrum are obtained by changing values of pairs $\{S(r),$

$S(\theta)\}$. These values are used as texture descriptors.

$$S(r) = \sum_{\theta=0}^{\pi} S_\theta(r) \tag{3.17}$$

$$S(\theta) = \sum_{r=1}^{R} S_r(\theta) \tag{3.18}$$

Wavelet Transforms

Wavelet transforms decompose a signal by means of a series of elementary functions, created from dilations and translations of a basis function ψ, known as mother wavelet. The basis functions of a discrete wavelet transform, $\psi_{j,k}(t)$, of time independent variable t, can be expressed as

$$\psi_{j,k}(t) = 2^{-j/2}\, \psi(2^{-j}t - k) \tag{3.19}$$

where j and k are integers that guide the dilations and translations of the function ψ to generate a family of wavelets, such as Haar and Daubechies [26, 66].

Wavelet transforms provide simultaneous time and frequency localization, whereas the standard Fourier transform is only localized in frequency. Additionally, wavelet transforms are useful for analyzing time-variant, non-stationary signals.

Using wavelets as a set of basis functions, an image can be decomposed into a multi-resolution hierarchy of localized information at different frequencies. The use of wavelet transform for texture analysis was first proposed by Mallat [66].

Wavelet transforms can be implemented by using a pair of low-pass and high-pass filters represented by a sequence of coefficients. In a 2D wavelet decomposition, the filters are applied to an image in both horizontal and vertical directions, typically followed by a downsampling. The output of each level will generate four subband images, LL, LH, HL and HH. The same process can be repeated on the LL image to generate the next decomposition level.

As wavelet coefficients in different frequency bands show variations in horizontal, vertical and diagonal directions, it has been shown that texture features can be extracted from these coefficients [119].

A well known feature based on wavelet coefficients is the energy, shown in Equation 3.20, where *sb* denotes the LL, LH, HL and HH subbands, $c(x, y)$ represents wavelet transform coefficients in the coordinates (x, y) for each one of these subbands containing $m \times m$ pixels. Wavelet energy reflects the distribution of energy along the frequency axis over scale and orientation and have proven to be very useful for texture characterization.

$$E_{sb} = \sqrt{\frac{1}{m^2} \sum c(x, y)^2} \tag{3.20}$$

Gabor Filters

Gabor filters capture visual properties such as spatial localization, spatial frequency and orientation of the structures present in the image. Widely employed to object recognition, Gabor filters present illumination invariance since they detect invariant spatial frequency [27]. The most common form of the Gabor filters is shown in Equation 3.21, where μ and ν denote the orientation and scale of the Gabor kernels, $z = (x, y)$, $\|\cdot\|$ is the norm operator, and $k_{\mu,\nu} = k_\nu(cos\phi_\mu, sin\phi_\mu)$, in which $k_\nu = k_{max}/f^\nu$ and $\phi_\mu = \pi\mu/8$ where k_{max} it the maximum frequency and f denotes the spacing factor between kernels in the frequency domain.

$$\psi_{\mu,\nu}(z) = \frac{\|k_{\mu,\nu}\|^2}{\sigma^2} e^{(-\|k_{\mu,\nu}\|^2\|z\|^2/2\sigma^2)}[e^{ik_{\mu\nu}z} - e^{-\sigma^2/2}] \tag{3.21}$$

The feature vector extracted using the Gabor filters is obtained with the convolution of the gray-scale image with the filters. Let $I(x, y)$ be the image, its convolution with a Gabor filter is defined according to Equation 3.22, where $*$ denotes the convolution operator.

$$G_{\psi I}(x, y, \mu, \nu) = I(x, y) * \psi_{\mu\nu}(z) \tag{3.22}$$

In this work, we consider five scales $\mu \in \{0, ..., 4\}$ and eight orientations $\nu \in \{0, ..., 7\}$, which results in 40 Gabor filters. For each filter, the image is convolved, generating 40 magnitudes. The mean and standard deviation of each image are calculated, resulting in 80 different features for the Gabor filters.

3.2.3 Geometrical Approach

In the geometrical approach, a texture is defined as being composed of primitives, also known as textels. After identifying primitives composing the texture, two classes of methods can be considered for feature extraction. The first uses descriptors extracted from the primitives to describe the texture, while the second considers rules to describe the spatial disposition of these primitives. The latter methods are referred to as structural and provide symbolic description of the texture. However, structural methods are not robust for noisy data. Therefore, we focus only on methods that extract descriptors from primitives.

Texture Unit

Considering that a texture can be seen as a set of small essential units able to characterize local information, He and Wang [42, 43] proposed the concept of texture unit, where measures computed from all units present in the texture can reveal its global aspects.

Given a 3×3 neighborhood composed by elements $V = \{v_0, v_1, \ldots, v_8\}$ where v_0 represents the intensity at the central pixel and the remaining v_i represents the intensity in its neighbors, texture unit is defined as the set $\mathrm{TU} = \{e_1, e_2, \ldots, e_8\}$, where each e_i is defined as

$$e_i = \begin{cases} 0, \text{ if } v_i < v_0 \\ 1, \text{ if } v_i = v_0 \\ 2, \text{ if } v_i > v_0 \end{cases} \qquad (i = 1, 2, \ldots, 8) \tag{3.23}$$

A signature, called texture spectrum, is obtained from the texture units. Each texture unit is associated with an index, obtained by Equation 3.24. There are 6561 ($= 3^8$) possible indices. The texture spectrum is a histogram with 6561 bins and the number of entries in the i-th bin, denoted by $S(i)$, is the number of texture units presenting the i-th index.

$$\mathrm{N_{TU}} = \sum_{i=1}^{8} 3^{i-1} e_i \tag{3.24}$$

Equation 3.24 defines how the index of the texture unit is computed, however, it does not specify an ordering for pixels v_i, for $i > 0$. To solve that, we consider that pixels are ordered in a clockwise order starting between a and h, as shown in Figure 3.2. Therefore, there are eight possible orders to compute $\mathrm{N_{TU}}$.

a	b	c
h		d
g	f	e

Figure 3.2: Neighborhood considered to compute texture unit.

He and Wang [42, 43] proposed a set of descriptors to be extracted from the texture spectrum. The black-white symmetry (BWS) is defined as

$$\mathrm{BWS} = 100 \left[1 - \frac{\sum_{i=0}^{3279} |S(i) - S(3281 + i)|}{\sum_{i=0}^{6560} S(i)} \right] \tag{3.25}$$

The descriptor called geometric symmetry (GS) computes information regarding regularity in the texture. It measures the symmetry between frequency of regions a and e, b and f, c and g, d and h, defined in Figure 3.2. GS is defined by Equation 3.26, where $S_j(i)$ contains the frequency of units with i-the index under ordering j, and j denotes one

of the possible orderings used to compute the texture unit.

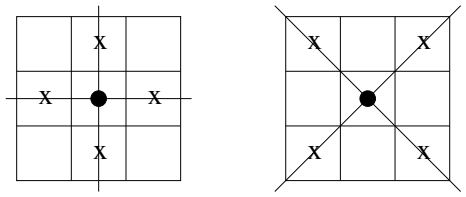
$$\mathrm{GS} = 100\left[1 - \frac{1}{4}\sum_{j=1}^{4}\frac{\sum_{i=0}^{6560}|S_j(i) - S_{j+4}(i)|}{2\sum_{i=0}^{6560}S_j(i)}\right] \tag{3.26}$$

Another descriptor, called degree of direction (DD), measures the degree of linearity of the primitives composing a texture. DD is defined as

$$\mathrm{DD} = 100\left[1 - \frac{1}{6}\sum_{m=1}^{3}\sum_{n=m+1}^{4}\frac{\sum_{i=0}^{6560}|S_m(i) - S_n(i)|}{2\sum_{i=0}^{6560}S_m(i)}\right] \tag{3.27}$$

Texture Feature Coding Method

Proposed by Horng et al. [45], the texture feature coding method (TFCM) is based on the connectivity of neighbors. First-order and second-order connectivities are considered, as defined in Figure 3.3.



(a) (b)

Figure 3.3: Connectivity defined by Horng et al. [45]. (a) first-order; (b) second-order.

TFCM results in pairs of integers (α, β), where α and β compute intensity changes in the first and second-order connectivities, respectively. The changes are computed by sweeping three adjacent pixels, according to a given connectivity. Let (a, b, c) be an ordering for a sweep and (g_a, g_b, g_c) be the intensity of these pixels. If two variations between adjacent pairs is considered, there are four possible types of sweep, as defined in Equation 3.28, where Δ is a threshold.

$$\begin{gathered}\text{(i) if } (|g_a - g_b| \leq \Delta) \text{ and } (|g_b - g_c| \leq \Delta) \\ \text{(ii) if } \Big[(|g_a - g_b| \leq \Delta) \text{ and } (|g_b - g_c| \geq \Delta)\Big] \text{ or} \\ \Big[(|g_a - g_b| \geq \Delta) \text{ and } (|g_b - g_c| \leq \Delta)\Big]\end{gathered}$$

$$\begin{gathered}\text{(iii) if } \Big[(g_a - g_b > \Delta) \text{ and } (g_b - g_c > \Delta)\Big] \text{ or} \\ \Big[(g_b - g_a > \Delta) \text{ and } (g_c - g_b > \Delta)\Big] \\ \text{(iv) if } \Big[(g_a - g_b > \Delta) \text{ and } (g_c - g_b > \Delta)\Big] \text{ or} \\ \Big[(g_b - g_a > \Delta) \text{ and } (g_b - g_c > \Delta)\Big]\end{gathered} \tag{3.28}$$

Since each order of connectivity presents two sweeps, Figure 3.4 can be used to summarize each sweep according to the order of connectivity. Given these possible values for α and β, the texture feature number is defined as $TFN(x, y) = \alpha(x, y)\beta(x, y)$, resulting in 41 different values.

	(i)	(ii)	(iii)	(iv)
(i)	1	2	3	4
(ii)	2	5	6	7
(iii)	3	6	8	9
(iv)	4	7	9	10

Figure 3.4: Values used to compute α and β. Rows indicate the type of the first sweep and columns the type of the second sweep.

Once the TFN is computed for each pixel in the texture, a histogram and a co-occurrence matrix (Section 3.2.1) are estimated. Both are used to compute the descriptors that compose the feature vector. The histogram is defined in Equation 3.29, where $N(i)$ denotes the frequency of the i-th TFN and n the number of pixels in the texture. The co-occurrence matrix is computed by Equation 3.30, where $N_{d,\theta}(i, j)$ denotes the number of transitions between TFN with value i to j given a distance d and orientation θ.

$$p_{\text{TFCM}}(i) = \frac{N(i)}{n}, \quad n \in \{0, 1, \ldots, 41\} \tag{3.29}$$

$$p_{\text{TFCM}}(i, j|d, \theta) = \frac{N_{d,\theta}(i, j)}{N_t}, \quad i, j \in \{0, 1, \ldots, 41\} \tag{3.30}$$

Based on the histogram, measures of coarseness (Co), homogeneity (Ho), mean convergence (MC) and variance (Va) are computed. These measures are defined as

$$\text{Co} = \sum_x \sum_y p_{\text{TFCM}}(41) \tag{3.31}$$

$$\text{Ho} = \sum_x \sum_y p_{\text{TFCM}}(0) \tag{3.32}$$

$$\text{MC} = \sum_{n=0}^{41} \frac{|n\, p_{\text{TFCM}}(n) - \mu|}{\sigma} \tag{3.33}$$

$$\mathrm{Va} = \sum_{n=0}^{41} (n-\mu)^2 \, p_{\mathrm{TFCM}}(n) \tag{3.34}$$

Using co-occurrence matrix measures of code entropy (CE) and code similarity (CS) are computed according to Equations 3.35 and 3.36.

$$\mathrm{CE} = -\sum_{i=0}^{41}\sum_{j=0}^{41} p_{\mathrm{TFCM}}(i,j|d,\theta) \, \log p_{\mathrm{TFCM}}(i,j|d,\theta) \tag{3.35}$$

$$\mathrm{CS} = \sum_{i=0}^{41} p^2_{\mathrm{TFCM}}(i,i|d,\theta) \tag{3.36}$$

Local Binary Patterns

Local Binary Patterns (LBP) characterizes the spatial structure of a texture and presents the characteristics of being invariant to monotonic transformations of the gray-levels [76]. On its standard version, a pixel c with intensity $g(c)$ is labeled as defined by Equation 3.37, where pixels p belong to a 3×3 neighborhood with gray levels g_p ($p = 0, 1, 2, ..., 7$).

$$S(g_p - g_c) = \begin{cases} 1, & g_p \geq g_c \\ 0, & g_p < g_c \end{cases} \tag{3.37}$$

Then, the LBP pattern of the pixel neighborhood is computed by summing the corresponding thresholded values $S(g_p - g_c)$ weighted by a binomial factor of 2^k as

$$\mathrm{LBP} = \sum_{k=0}^{7} S(g_p - g_c) 2^k \tag{3.38}$$

After computing the labeling for each pixel of the image, a 256-bin histogram of the resulting labels is used as a feature descriptor for the texture.

Several variations of LBP have been proposed, including the Improved Local Binary Pattern (ILBP) [52]. Different from the LBP, the ILBP uses the average of the 3×3 neighborhood as the threshold and also considers the central pixel to estimate its label, with values in the interval $[0, 510]$. The ILBP is defined as in Equation 3.39.

$$\mathrm{LBP} = \sum_{k=0}^{7} S(g_p - m) 2^k + S(g_c - m) 2^8 \tag{3.39}$$

where m denotes the average of the 3×3 neighborhood and S(x) is defined as

$$S(x) = \begin{cases} 1, & g_p > 0 \\ 0, & g_p \leq 0 \end{cases} \tag{3.40}$$

Similarly to LPB, once the labels have been computed for every pixel, an histogram is computed, in this case a 511-bin histogram, which will be used as feature vector.

Coordinated Clusters Representation

The coordinated clusters representation (CCR), proposed by Kurmyshev and Cervantes [57], is a descriptor for binary texture, where the image is characterized by a histogram of occurrence of the possible binary patterns. This descriptor was later extended to grayscale texture images [101].

A matrix of binary image intensities is denoted as $S^\alpha = \{s^\alpha l, m\}$, where $l = 1, 2, ..., L$, $m = 1, 2, ..., M$, $N = L \times M$ and $\alpha = 1, 2, ..., 2^N$ is the index of the image. The binary image S^α is scanned with one-pixel steps by a rectangular window $W = I \times J$ pixels. Since pixels correspond to binary units, the number of all possible states of the window is 2^W.

The pixel configuration found by the window is coded as the binary number corresponding to this configuration. The coordinated cluster representation for a given binary image S^α consists of a histogram $H^\alpha_{(I,J)}(b)$ of occurrence of the pixel patterns detected by the scanning window. When the histogram $H^\alpha_{(I,J)}(b)$ is normalized, it can be interpreted as a distribution function of occurrences given by

$$F^\alpha_{(I,J)}(b) = \frac{H^\alpha_{(I,J)}(b)}{A} \tag{3.41}$$

Granulometry

The term granulometry is used in the field of materials science to characterize the granularity of materials by passing them through sieves of different sizes while measuring their mass retained by each sieve.

This principle can be transposed to the field of image processing [39, 44, 104], where an operator consists in analyzing the amount of image detail removed by applying morphological openings γ_λ of increasing size λ.

The mass is represented by the sum of the pixel values, known as image volume (Vol). The volumes of the opened images are plotted against λ, producing a granulometric curve. The normalized version of the operator for an image f can be written as

$$G(f) = \frac{\text{Vol}(\gamma_\lambda(f))}{\text{Vol}(f)} \tag{3.42}$$

Negative values of λ can be interpreted as a morphological closing operator with a structuring element of size λ.

3.2.4 Approach Based on Parametric Models

In this approach, a texture is considered as a sample from a stochastic process defined by a set of parameters, which are used to summarize the texture. Analysis and synthesis of texture can be performed with the use of these parameters. This section describes parametric models based on Markov random fields [11,37] and simultaneous autoregressive models [68, 98].

Markov Random Fields

Texture analysis based on Markov random fields (MRF) uses the set of parameters estimated from the probability distribution as descriptors. The model of local descriptors proposed by Ising [50] was used by Cross and Jain [22] to model textures considering neighborhoods of first, second, third, and fourth orders. The Ising model, originally proposed to empirically explain observed facts about ferromagnetic materials, is an important and simple representation in MRF, where a lattice of points can be used to model the interaction of neighboring points.

Local descriptors are defined by Equation 3.43, where parameters q depend on the neighborhood order, which is defined according to the scheme shown in Figure 3.5.

$$P(X_s = x_s|\partial_s) = \frac{\exp(-qx)}{1 + \exp(-q)} \tag{3.43}$$

where $S = \{1, 2, .., n\}$, X_s is a random variable, and ∂_s is a collection of neighbors in S.

	o_1	m	q_1	
o_2	v	u	z	q_2
l	t	s	t'	l'
q'_1	z'	u'	v'	o'_1
	q'_2	m'	o'_2	

Figure 3.5: Representation of the neighborhood of a central pixel with intensity x_s.

The first order neighborhood model uses q defined as

$$\mathrm{q}_1 = a + b_{1,1}(x_t + x_{t'}) + b_{1,2}(x_u + x_{u'}) \tag{3.44}$$

the second order neighborhood

$$\mathrm{q}_2 = q_1 + b_{2,1}(x_v + x_{v'}) + b_{2,2}(x_z + x_{z'}) \tag{3.45}$$

the third order neighborhood

$$\mathrm{q}_3 = q_2 + b_{3,1}(x_m + x_{m'}) + b_{3,2}(x_l + x_{l'}) \tag{3.46}$$

and, finally, for the fourth order

$$\mathrm{q}_4 = q_3 + b_{4,1}(x_{o_1} + x_{o'_1} + x_{o_2} + x_{o'_2}) + b_{4,2}(x_{q_1} + x_{q'_1} + x_{q_2} + x_{q'_2}) \tag{3.47}$$

Coefficients a and $b_{i,j}$ are used as texture descriptors. Their estimation is performed using the coding method proposed in [11].

Simultaneous Autoregressive Models

As an instance of the MRF models, the simultaneous autoregressive models (SAR) have been successfully applied to texture classification by considering the spatial interactions among neighboring pixels to represent textures [68]. Therefore, similarly to MRF, a SAR model for a pixel is defined as a function of its neighbors, as shown in Equation 3.48, where $f(x, y)$ is the image intensity at location (x, y), N defines its neighborhood, $\theta(k, l)$ are the model parameters and $E(m, n)$ is an error term associated to the pixel.

$$f(x, y) = \sum_{(k,l)\in N} \theta(k, l) f(m - k, n - l) + E(m, n) \tag{3.48}$$

Given a texture, parameters $\theta(k, l)$ are estimated considering small neighborhoods around each pixel and, in this work the parameter estimation is performed using the Least Square Error (LSE) technique. Then, the concatenation of these parameters composes the feature vector used to describe the texture.

3.2.5 Summary of Texture Descriptors

Table 3.1 presents a summary of the texture descriptors described in the previous sections. Main parameters and number of dimensions are shown for each texture descriptor categorized into its respective approach.

Approach	Descriptor	Parameters	Reference (Section)	Dimension
Statistical: *texture properties are represented in indirect and probabilistic manners.*	First order Statistics	-	3.2.1	6
	GLCM	orientations, distance between two pixels and quantization levels	3.2.1	48
	GLRLM	orientations	3.2.1	10
	Autocorrelation	-	3.2.1	81
Signal Processing: *image transforms are applied to extract feature descriptors.*	Fourier Spectrum	-	3.2.2	2
	Wavelets	wavelet basis	3.2.2	6
	Gabor Filters	number of scales and orientations	3.2.2	80
Geometrical: *descriptors are extracted from texture primitives (textels) initially identified.*	Texture Unit	-	3.2.3	3
	TFCM	Δ, GLCM displacement	3.2.3	6
	LBP	-	3.2.3	256
	ILBP	-	3.2.3	512
	CCR	window size	3.2.3	16, 256, 65536
	Granulometry	structuring element size, step to increase the structuring element	3.2.3	24-104
Parametric Models: *texture is considered as a sample from a stochastic process defined by a set of parameters used to describe the texture.*	MRF	order of the neighborhood	3.2.4	3-9
	SAR	size of the neighborhood	3.2.4	9

Table 3.1: Summary of feature descriptors classified according to their approaches.

3.3 Experimental Results

This section describes the experiments conducted to evaluate the feature descriptors applied to texture classification. We consider four texture data sets: UIUC [60], UMD [118], Outex [74] and VisTex [121]. All data sets were used to estimate and evaluate the parameters employed to assess the effectiveness of the feature descriptors for texture classification.

In Section 3.3.1, we describe the setup used for each method. Section 3.3.2 presents a brief explanation of each data set used in our experiments. Then, the feature extraction methods are compared and discussed in Section 3.3.3.

3.3.1 Experimental Setup

Details of implementation for the feature extraction methods as well as the parameters used by each are given as follows.

Markov Random Fields. We consider first, second, third, and fourth neighborhood orders for the generalized Ising model used for the MRF, having 3, 5, 7, and 9 parameters, respectively.

Autocorrelation. The implementation of the autocorrelation function possesses two parameters to be estimated, p and q. These parameters are directly related to the number of variables in the feature vector. Their optimum values are experimentally estimated. According to the experiments, parameters $p = q = 9$ achieved the best results and will be considered in the remaining experiments.

Gray Level Co-occurrence Matrix. A subset of the 14 descriptors described in Section 3.2.1 is considered for the co-occurrence matrices: angular second moment, contrast, correlation, sum of squares, inverse difference moment, sum average, sum variance, sum entropy, entropy, difference variance, difference entropy, and maximal correlation coefficient. Co-occurrence matrices for four orientations are computed (0^o, 45^o, 90^o, and 135^o) and used to compose the feature vector. The optimum value for parameter d is experimentally estimated to be 1 and four orientations are considered and concatenated, resulting a feature vector with 48 dimensions.

Gray Level Run Length Matrices. Regarding the gray level run length matrices, we consider all descriptors presented in Section 3.2.1 for orientation of 0^o and 90^o, therefore, resulting in a feature vector composed of 10 variables.

Coordinated Clusters Representation. The CCR method has the window size as free parameter.

Granulometry. The parameters considered for the granulometry are the kernel size, using morphological opening and closing operations, and the step parameter, which indicates the increment in size at a time.

Texture Feature Coding Method. Experiments were performed to estimate the best value for the parameters Δ and the displacement of the co-occurrence matrix, described in Section 3.2.3.

Wavelets. We used the Daubechies wavelet basis with two levels of decomposition in the wavelet-based method. The energy coefficients are obtained from the sub-images with high frequency. Therefore, the feature vector has 6 variables. In the experiments, we considered two wavelet bases. The inputs are necessarily square images, otherwise they will be cropped to the largest possible square.

Fourier Spectrum. As a restriction of this method, the inputs must be power of 2 grayscale images, otherwise they will be cropped.

Simultaneous Autoregressive Models. Parameters k_1 and k_2 are used to indicate the size of the neighborhood. The experiments were conducted using $k_1 = k_2$, varying k_1 between 6 to 10. The feature vector has 9 dimensions.

3.3.2 Data Sets

This section describes the main characteristics of the four data sets used in our experiments.

UMD Data Set

The UMD high-resolution data set [118] contains images of 1280×960 pixels, with 1000 images split into 25 classes, giving a total of 40 samples per class. This data set includes images of floor textures, plants, fruits, among others. A mosaic containing examples of all classes can be seen in Figure 3.6.

UIUC Data Set

The UIUC data set [60] is a texture data set composed of 1000 images of 640×480 pixels, distributed in 25 classes (variations of wood, gravel, fur, carpet, brick, among others) with 40 samples each. Figure 3.7 shows examples of images for each texture.

OuTex Data Set

OuTex [78] is a framework for evaluation of texture classification and segmentation. It contains several images and protocols for texture classification. The images are acquired under several illuminations and different angles, over a number of different textures.

Several test suites for texture classification have been proposed. Out of all these, TC_00005 is used in this work, since its results are not saturated. This test suite contains

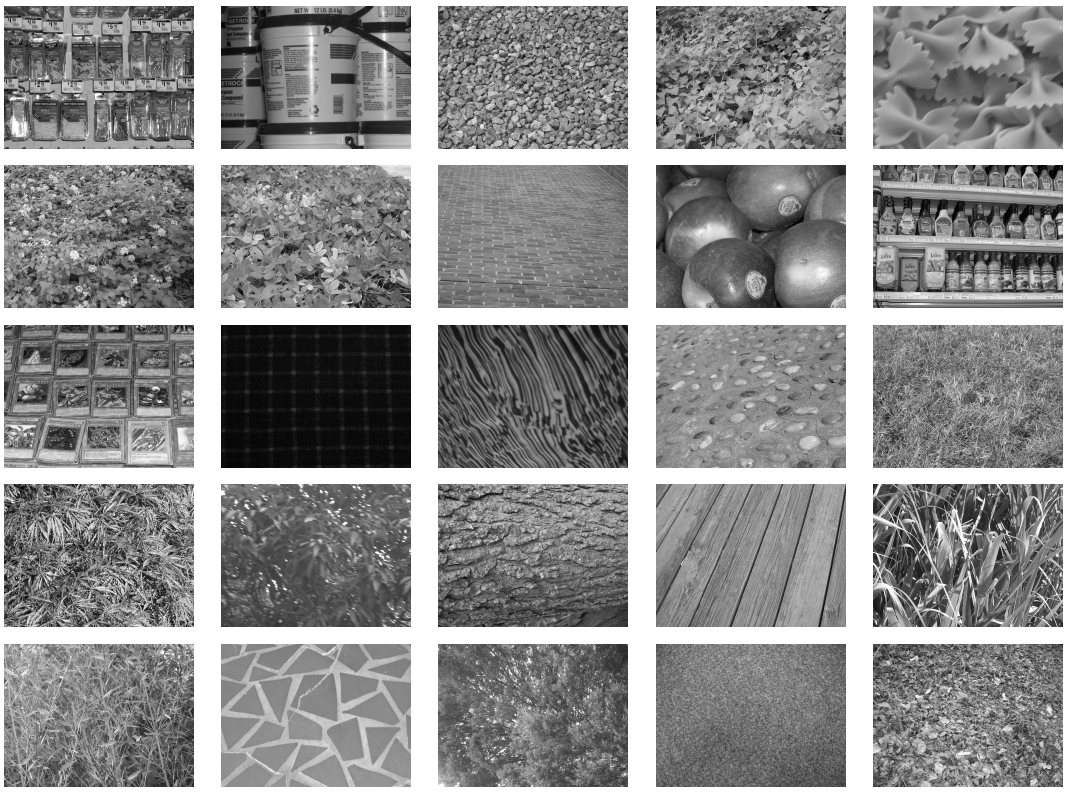


Figure 3.6: Examples of 25 texture samples extracted from UMD data set [118].

8832 sample images of 32 × 32 pixels, belonging to 24 classes of textures, and a hundred of different arranges for training and testing. Figure 3.8 shows examples of texture images.

VisTex Data Set

VisTex [121] is a collection of texture images that are representative of real world conditions. Our experiments included 54 images of resolution with 512 × 512 pixels split into 16 samples of 128 × 128 pixels, according to work described by Arvis et al. [5]. Such images are available on the Outex site [78] as test suite Contrib_TC_00006. For each texture class, half of the samples were used in the training set and the other half were used as testing data. Figure 3.9 shows examples of texture images.

3.3.3 Results and Comparisons

To perform texture classification, each data set was partitioned into two sets: training and test. One hundred random splits were employed for UMD and UIUC datasets, considering five, ten, fifteen and twenty training samples, whereas the remaining samples

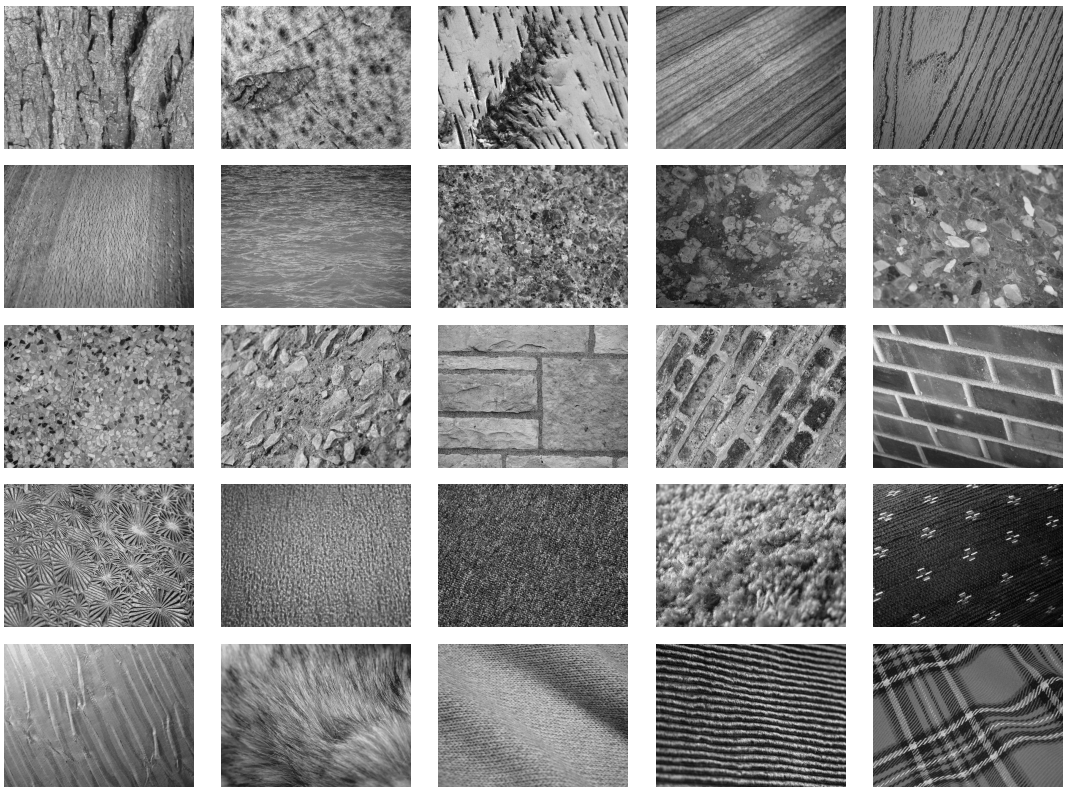


Figure 3.7: Examples of 25 texture samples extracted from UIUC data set [60].

were used as test. For OuTex and VisTex data sets, the test suites TC_00005 [78] and Contrib_TC_00006 [121] were considered in our experiments, respectively.

The reported results correspond to the average of all the considered splits. Nearest neighbor (1-NN) classifier is applied after all variables are normalized to present zero mean and unit variance in order to reduce the predominance of certain feature values. It is worth pointing out that there was no contamination between training and test patterns in the experiments. The parameter estimation for the normalization (mean and variance) is performed by using only patterns belonging to the training set.

Dimensionality is reduced by using Principal Component Analysis (PCA). Since 1-NN is used as classifier and it relies on the distance between feature descriptors extracted from the patterns, the direct use of the classifier without reducing the dimensionality of the data with PCA would suffer from the curse of dimensionality, in which the feature space becomes sparser as the dimensionality increases.

Four distance functions are considered to measure the similarity between pairs of samples: Euclidean, city block, cosine, and correlation. Besides the results achieved using single feature extraction methods, results considering feature combination are also presented. The combination of the features is constructed by concatenating all features

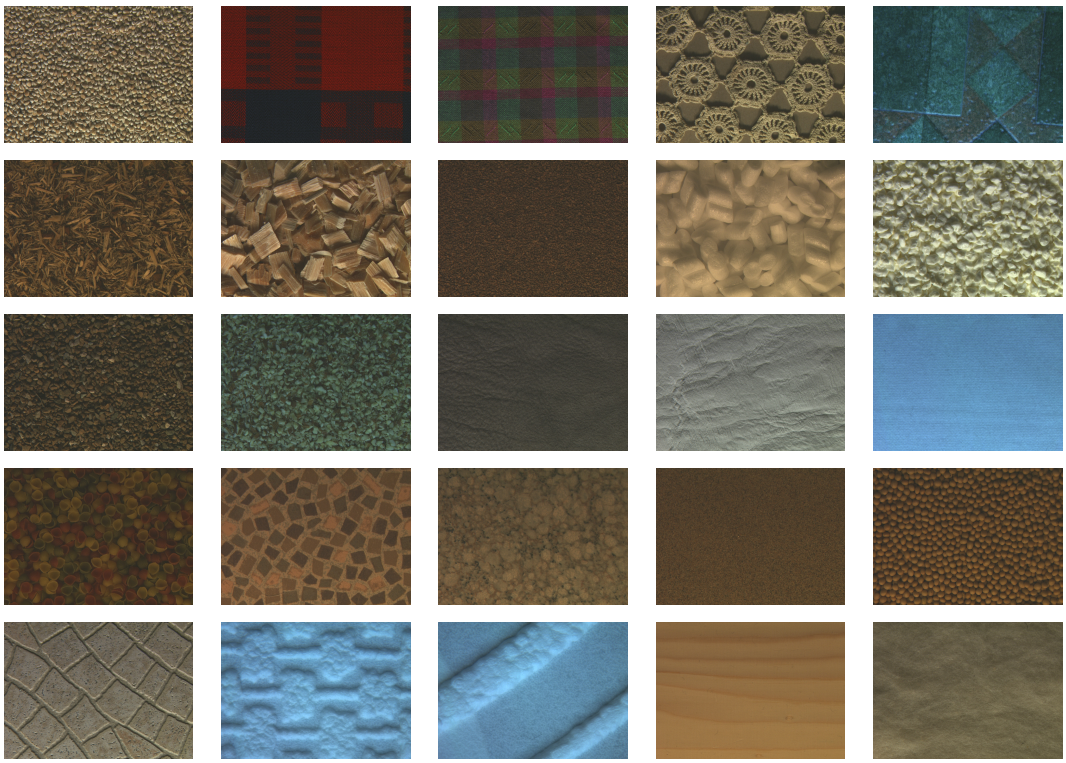


Figure 3.8: Examples of texture samples extracted from OuTex data set [78].

together.

Tables 3.2 to 3.5 show the classification rates for each data set considering different feature parameters. The best results achieved for each feature extraction method are displayed in Figure 3.10. This figure summarizes the results for all data sets showing in the x-axis normalized results (the closer to 100, the better the results achieved by a given feature extraction method). Methods ILBP, LBP, Gabor, GLCM and granulometry are the methods that achieved the best results throughout all experiments performed. Finally, it is important to point out that the feature combination achieved the highest texture classification rates for all data sets.

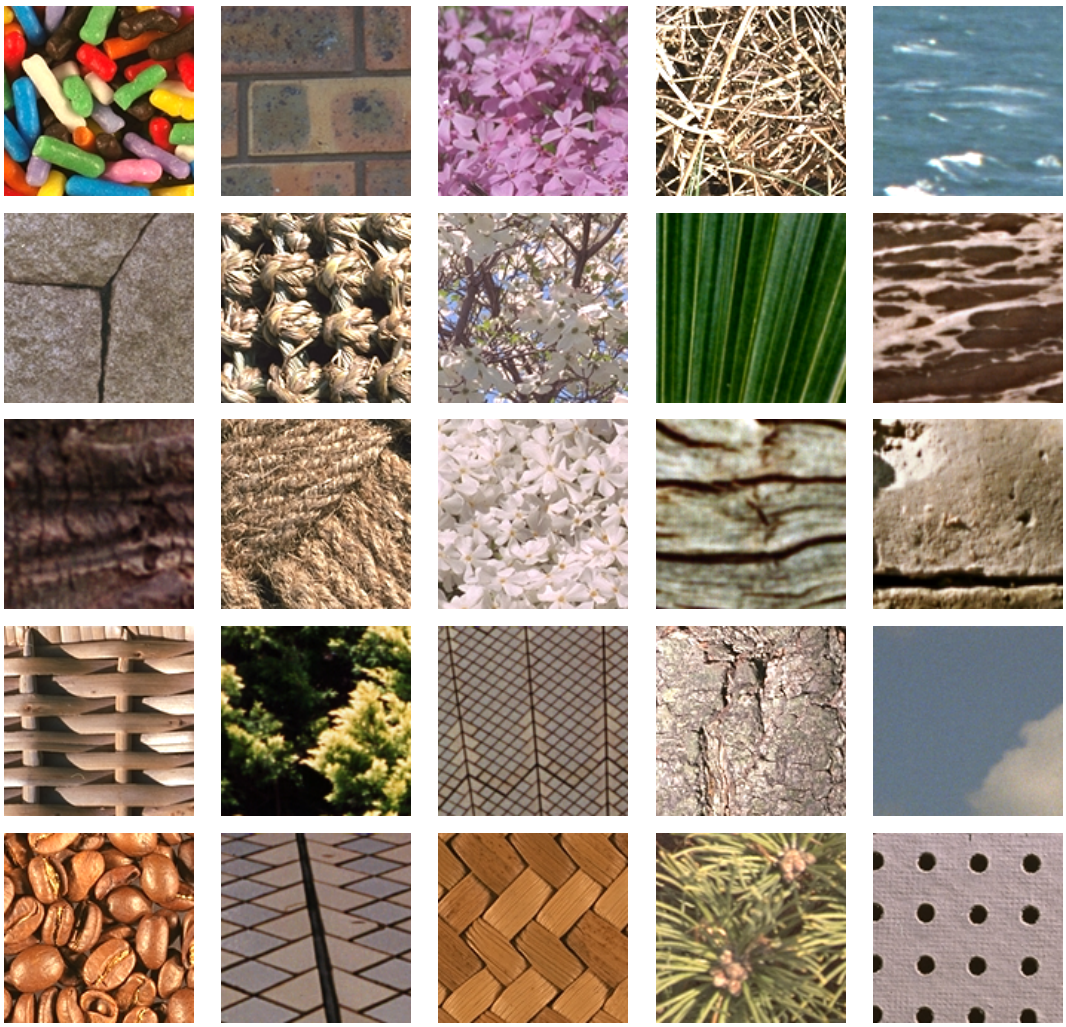


Figure 3.9: Examples of texture samples extracted from Vistex data set [121].

Features	Parameters			Classification Rates (%)			
				5	10	15	20
First order statistics	-			50.272	53.582	55.328	56.256
GLCM	Θ	d	level				
	0°, 45°, 90°, 135°	1	256	82.178	87.898	90.524	92.132
GLRLM	Θ						
	0° and 90°			62.068	69.420	72.603	74.698
Autocorrelation	-			55.772	64.317	69.553	72.676
Fourier spectrum	-			18.800	18.689	18.692	18.492
Wavelets	basis						
	db4			53.088	61.734	66.105	69.526
	db8			55.080	62.633	66.464	69.146
Gabor filters	scale		Θ				
	5		8	75.695	82.329	84.857	88.098
Texture unit	-			34.859	36.777	38.217	39.168
TFCM	Δ		d				
	0		1	55.420	61.885	64.995	67.516
	0		2	56.600	63.110	66.467	69.174
	0		3	56.648	63.256	66.563	69.370
	0		4	56.502	62.750	66.300	68.792
	0		5	55.818	62.301	65.545	68.026
	1		1	59.065	66.406	69.968	72.472
	1		2	59.006	66.965	70.808	73.762
	1		3	58.524	66.562	70.227	73.356
	1		4	58.126	66.166	70.134	73.346
	1		5	57.658	65.576	69.438	72.482
LBP	-			72.522	81.501	85.468	88.188
ILBP	-			73.356	81.366	85.081	87.432
CCR	window						
	3			68.996	77.509	81.756	84.276
	4			68.794	77.494	81.800	84.334
Granulometry	kernel size		step				
	5		1	69.996	79.114	83.956	86.372
	5		2	70.609	79.756	84.515	87.124
	10		1	73.461	82.340	87.035	89.504
	10		2	73.569	82.345	86.982	89.490
	15		1	73.601	82.468	87.201	89.840
	15		2	73.893	82.789	87.561	90.140
	20		2	74.386	83.224	87.708	90.246
	25		2	75.435	83.861	88.076	90.698
MRF	order						
	0			30.845	33.246	34.809	35.894
	1			48.506	55.076	60.632	63.366
	2			57.272	65.117	70.748	74.184
	3			61.336	69.560	74.347	77.738
SAR	size						
	6			65.298	73.606	77.660	80.308
	7			64.082	72.486	76.825	79.342
	8			62.225	70.805	75.508	77.902
	9			60.219	69.188	74.096	76.780
	10			58.030	67.402	72.296	75.168
Feature combination	-			85.010	89.448	85.006	92.080

Table 3.2: Results of the experiments on the UMD data set as function of the number of training samples.

Features	Parameters			Classification Rates (%)			
				5	10	15	20
First order statistics	-			40.536	44.465	44.108	48.548
GLCM	Θ	d	level				
	0°, 45°, 90°, 135°	1	256	53.817	62.269	66.761	69.642
GLRLM	Θ						
	0° and 90°			26.180	28.850	30.777	31.804
Autocorrelation	-			29.893	35.220	38.752	40.988
Fourier spectrum	-			13.750	13.445	13.475	13.400
Wavelets	basis						
	db4			26.216	30.341	32.979	34.392
	db8			25.229	28.833	31.216	36.172
Gabor filters	scale		Θ				
	5		8	50.961	59.352	65.110	69.861
Texture unit	-			22.948	24.780	24.899	25.328
TFCM	Δ		d				
	0		1	24.461	27.244	28.484	29.060
	0		2	24.340	27.032	28.123	28.552
	0		3	24.667	27.521	28.827	29.296
	0		4	25.056	27.872	29.505	29.956
	0		5	25.012	27.797	29.377	29.980
	1		1	27.342	31.424	33.038	34.446
	1		2	27.157	31.380	33.168	34.462
	1		3	27.248	31.549	33.372	34.900
	1		4	27.429	31.552	33.558	34.948
	1		5	27.419	31.656	33.673	35.056
LBP	-			41.712	44.681	49.272	53.489
ILBP	-			44.808	51.737	54.765	61.327
CCR	window						
	3			34.427	40.752	44.361	46.748
	4			34.420	40.774	44.388	46.802
Granulometry	kernel size		step				
	5		1	40.934	50.266	55.366	59.238
	5		2	41.737	50.701	55.651	59.210
	10		1	44.481	54.433	59.750	63.980
	10		2	44.556	54.321	59.691	63.746
	15		1	45.594	56.154	61.604	65.610
	15		2	46.196	56.933	62.395	66.306
	20		2	47.024	57.821	63.276	67.252
	25		2	48.353	58.924	64.592	68.674
MRF	order						
	0			19.356	20.448	20.780	21.274
	1			27.438	30.637	32.484	34.284
	2			30.004	35.129	38.057	40.204
	3			28.581	33.844	36.851	39.032
SAR	size						
	6			28.217	33.912	37.190	39.842
	7			26.138	31.278	34.054	36.672
	8			23.667	28.228	30.715	33.254
	9			21.825	25.605	27.838	30.184
	10			20.158	23.432	25.054	27.072
Feature combination	-			54.807	63.921	68.683	71.842

Table 3.3: Results of the experiments on the UIUC data se as function of the number of training samples.

Features	Parameters			Classification Rates (%)
First order statistics	-			44.675
GLCM	Θ	d	level	
	0°, 45°, 90°, 135°	1	256	94.444
GLRLM	Θ			
	0° and 90°			63.888
Autocorrelation	-			76.620
Fourier spectrum	-			14.814
Wavelets	basis			
	db4			78.703
	db8			80.092
Gabor filters	scale		Θ	
	5		8	92.592
Texture unit	-			32.638
TFCM	Δ		d	
	0		1	67.129
	0		2	64.583
	0		3	62.037
	0		4	63.194
	0		5	62.735
	1		1	72.685
	1		2	71.759
	1		3	71.759
	1		4	68.750
	1		5	71.527
LBP	-			96.527
ILBP	-			97.916
CCR	window			
	3			80.787
	4			80.787
Granulometry	kernel size		step	
	5		1	93.518
	5		2	93.750
	10		1	94.444
	10		2	93.518
	15		1	91.203
	15		2	91.666
	20		2	91.435
	25		2	88.657
MRF	order			
	0			30.324
	1			61.574
	2			74.074
	3			83.564
SAR	size			
	6			80.787
	7			71.990
	8			66.435
	9			61.111
	10			58.564
Feature combination	-			99.768

Table 3.4: Results of the experiments on the VisTex data set.

Features	Parameters			Classification Rates (%)
First order statistics	-			51.205
GLCM	Θ	d	level	
	0°, 45°, 90°, 135°	1	256	83.147
GLRLM	Θ			
	0° and 90°			59.703
Autocorrelation	-			40.265
Fourier spectrum	-			10.441
Wavelets	basis			
	db4			52.513
	db8			49.181
Gabor filters	scale		Θ	
	5		8	84.912
Texture unit	-			25.995
TFCM	Δ		d	
	0		1	44.789
	0		2	42.053
	0		3	42.121
	0		4	39.188
	0		5	37.862
	1		1	43.152
	1		2	40.591
	1		3	39.471
	1		4	37.242
	1		5	37.135
LBP	-			89.044
ILBP	-			82.091
CCR	window			
	3			69.353
	4			69.523
Granulometry	kernel size		step	
	5		1	89.266
	5		2	87.428
	10		1	87.376
	10		2	87.007
	15		1	86.723
	15		2	86.576
	20		2	85.139
	25		2	83.713
MRF	order			
	0			20.692
	1			30.857
	2			41.658
	3			42.629
SAR	size			
	6			49.555
	7			45.338
	8			41.378
	9			35.777
	10			31.605
Feature combination	-			93.922

Table 3.5: Results of the experiments on the OuTex data set.

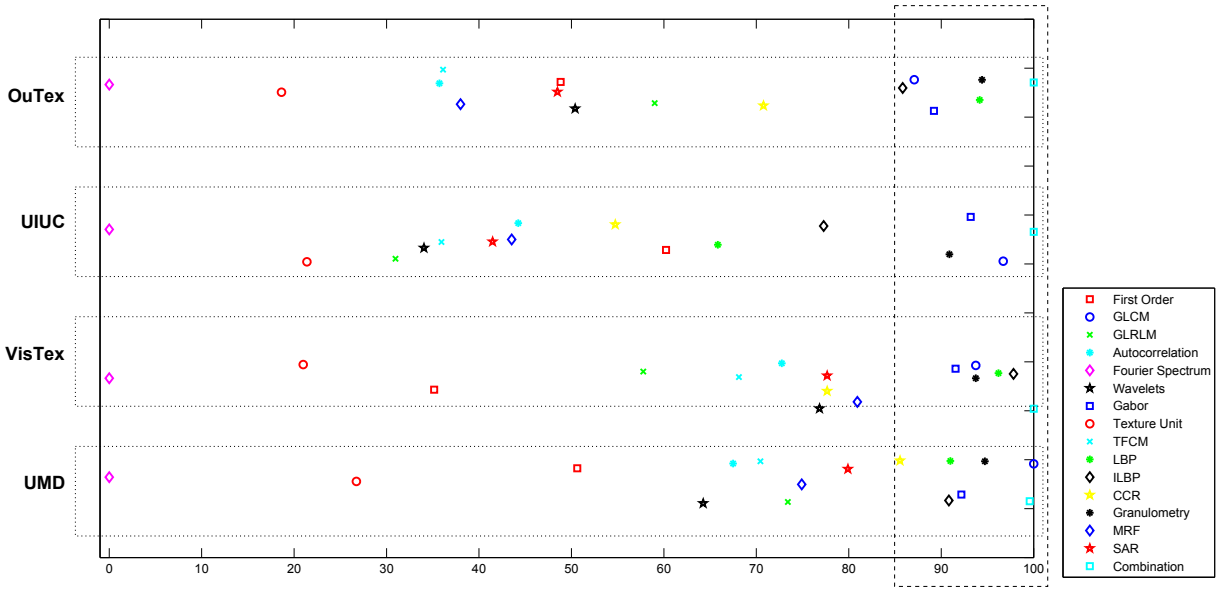


Figure 3.10: Comparison of results for all considered features on each data set. Each horizontal dotted box displays results for one data set. The x-axis shows normalized results (between 0 and 100) according to the minimum and maximum classification rates achieved for each data set. The vertical slashed box on the right-hand side selects the feature extraction methods that achieved the best results throughout all data sets.

3.4 Conclusions

Texture information plays an important role in several domains, such as image segmentation, image classification, content-based image retrieval, scene recognition, video analysis, among others. Although several texture descriptors have been proposed in the literature, the formal definition of texture in images is still a complex task.

This work presented and compared a set of texture descriptors to assess their suitability for texture classification. Statistical, signal processing, geometrical, and parametric model approaches are used to extract texture features from the images.

Experiments demonstrated that feature descriptors such as ILBP, LBP, Gabor filters, GLCM and granulometry present consistently high classification rates in different data sets. Furthermore, the combination of features, formed by the concatenation of them together, produced superior results in terms of classification rate when compared to the use of the features individually.

Chapter 4

Multi-Scale Gray Level Co-Occurrence Matrices for Texture Description

Preamble

Texture information plays an important role in image analysis. Although several descriptors have been proposed to extract and analyze texture, the development of automatic systems for image interpretation and object recognition is a difficult task due to the complex aspects of texture. Scale is an important information in texture analysis, since a same texture can be perceived as different texture patterns at distinct scales. Gray level co-occurrence matrices (GLCM) have been proved to be an effective texture descriptor. This paper presents a novel strategy for extending the GLCM to multiple scales through two different approaches, a Gaussian scale-space representation, which is constructed by smoothing the image with larger and larger low-pass filters producing a set of smoothed versions of the original image, and an image pyramid, which is defined by sampling the image both in space and scale. The performance of the proposed approach is evaluated by applying the multi-scale descriptor on five benchmark texture data sets and the results are compared to other well-known texture operators, including the original GLCM, that even though faster than the proposed method, is significantly outperformed in accuracy.

4.1 Introduction

Texture can be characterized by regular or random patterns that repeat over a region [102]. As one of the most important features for image analysis, texture provides information regarding structural arrangement of surfaces or changes in intensity or color brightness.

Despite the accuracy of the visual human system to recognize textures, it is a complex task to define a set of textural descriptors for image analysis on different domains of knowledge. The large number of definitions and descriptors found in the literature reflects such difficulty [40, 59, 80–84, 99, 112, 113, 127].

Although there is no unique categorization of the main relevant methods for texture description, they can be classified as *statistical approaches*, *signal-processing based approaches*, *geometrical approaches*, and *parametric-model based approaches* [116].

Among the statistical approaches, gray level co-occurrence matrices (GLCM) have been proved to be a very powerful texture descriptor used in image analysis. However, a drawback of the original GLCM is its limited capability of capturing texture information at multiple scales.

In many descriptors, it is assumed that texture information is fixed at a specific image resolution. The discriminative power of texture descriptors can be significantly improved if different scales are considered among the images during the descriptor extraction.

This work presents a novel scheme for extending the GLCM to be more robust under scale variation. Two different multi-scale representations are used in the extension of the descriptor. The performance of the proposed approach is evaluated by applying the multi-scale descriptor on five benchmark texture data sets and the results are compared to other powerful texture operators.

The paper is organized as follows. Section 4.2 presents some relevant concepts and work related to texture descriptors. Section 4.3 describes the proposed multi-scale texture descriptor. Experimental results are shown in Section 4.4. Finally, Section 6 concludes the paper with final remarks.

4.2 Related Work

The development and analysis of low-level feature descriptors have been widely considered in the past years. Among the vastly employed methods are the scale-invariant feature transform (SIFT) [65], speeded up robust feature (SURF) [8], histogram of oriented gradients (HOG) [24], gradient location and orientation histogram (GLOH) [71], region covariance matrix (RCM) [117], edgelet [125], gray level co-occurrence matrix (GLCM) [41], local binary patterns (LBP) [76], color correlogram (CCG) [48], color coherence vectors (CCV) [85], color indexing [110], steerable filters [32], Gabor filters [51], and shape con-

text [9]. Furthermore, several works comparing different feature descriptors can be found in the literature [71, 98, 132].

Some descriptors have taken scale changes into account. For instance, extensions of LBP have been proposed to make it a more effective texture descriptor. One extension is the multi-resolution LBP (MLBP) [77], which uses pixel neighborhood of different sizes. Another extension represents LBP descriptors in Gabor transform domain (LGBP) [133]. A third variation extends LBP to pyramid transform domain (PLBP) [94]. PHOG [13] represents an image with histograms of orientation gradients over spatial pyramids. HWVP [95] represents texture information of an image through a hierarchical wavelet packet transform. GIST [114] is a high dimensional descriptor that represents texture information with filtering based on oriented multiple scale.

Low-level descriptors are designed to focus in visual characteristics such as texture, shape and color. Since this paper proposes a multi-scale extension of the texture-based method GLCM, we review approaches to extracting textural information. Section 4.2.1 describes a set of texture-based feature descriptors. Then, Section **??** reviews the GLCM method and its extensions.

4.2.1 Texture Descriptors

Many approaches to extracting textural information have been proposed in the literature, including gray level co-occurrence matrices [41], gray level run length matrices [34], wavelet transforms [119], Gabor filters [27], texture unit [42], local binary patterns [76], texture feature coding method [45], coordinated clusters representation [57], granulometry [39], Markov random fields [22], and simultaneous autoregressive models [68]. This section reviews those approaches that have achieved accurate results in texture classification [28, 98, 107, 131]. These descriptors are then compared to the proposed method during the experimental validation (Section 4.4).

As previously discussed in Chapter 3, there are many different texture descriptors that achieve high detection rate. In this work, LBP, GLCM, Gabor and Granulometry are used to compare the results against the proposed multi-scale extension of GLCM.

Several variations of the original version of the GLCM have been proposed. Focusing on optimization, Clausi and Jernigan [21] employ linked lists exploiting the sparsity of the co-occurrence matrices to reduce the computation time; Tahir et al. [111] propose the use of Field Programmable Gate Arrays (FPGA) to accelerate the calculation of GLCM and Haralick texture features, achieving speed-up of 9 times.

Extensions of the GLCM have been proposed. To increase the discriminability of the descriptors, Gelzinis et al. [36] extract descriptors considering simultaneously different values for parameter d. Walker et al. [122] have proposed to form co-occurrence matrix-

based features by weighted summation of GLCM elements from areas presenting high discrimination. Furthermore, addition of color information has also been considered for co-occurrence matrices [10].

Multi-scale analysis has also been performed using the GLCM. Hu [47] and Pacifici et al. [79] consider multiple scales by changing the window size from which the GLCM descriptors are extracted. Rakwatin et al. [97] propose that the image be rescaled to different sizes, extracting co-occurrence descriptors from each size. Nguyen-Duc et al. [72] have obtained improved results on content-based image retrieval employing a combination of contourlet transform [29] and GLCM. First, the contourlet transform is performed for four subbands of the image, then the GLCM features are extracted from each one.

Differently from previous works that have addressed multiple scales for GLCM, this paper exploits two approaches to performing the multi-scale analysis on the images: pyramid decomposition and Gaussian smoothing. Furthermore, five strategies to combine the descriptors extracted from each scale are considered and evaluated.

4.3 Multi-Scale Gray Level Co-Occurrence Matrix

Two common multi-resolution representations are Gaussian smoothing and pyramid decomposition [64]. Gaussian smoothing is a space-scale representation, where a sequence of images at different levels of space-scale are built with variable kernel sizes. In pyramid representation, the original image is progressively reduced at each level of the pyramid. Pyramid images can be generated by applying the Gaussian smooth filtering, Laplacian operator, low-pass filters of wavelet transform, among other schemes.

The use of multiple scales can improve the discriminative power of texture descriptors, since they are able to extract information that could not be present in certain scales. This work proposes an extension of the gray level co-occurrence matrix (GLCM) to multiple scales using the Gaussian smoothing and pyramid decomposition to generate a number of image representations of the original image.

The co-occurrence matrix is created at each level of resolution, resulting in several scales. Features extracted from GLCM are then combined into a single feature descriptor. All features are extracted from images in grayscale, that is, color data sets are converted into grayscale before their extraction. Figure 4.1 illustrates the main steps of the multi-scale approaches.

Once the features are extracted from each scale, it is necessary to combine them in a manner to take advantage of the multiple scale information. To achieve that, five different combination strategies are proposed to merge the features.

The first one, referred to as *concatenation*, is a simple union of the scales, that is, features extracted from each scale are concatenated in a single feature vector that contains

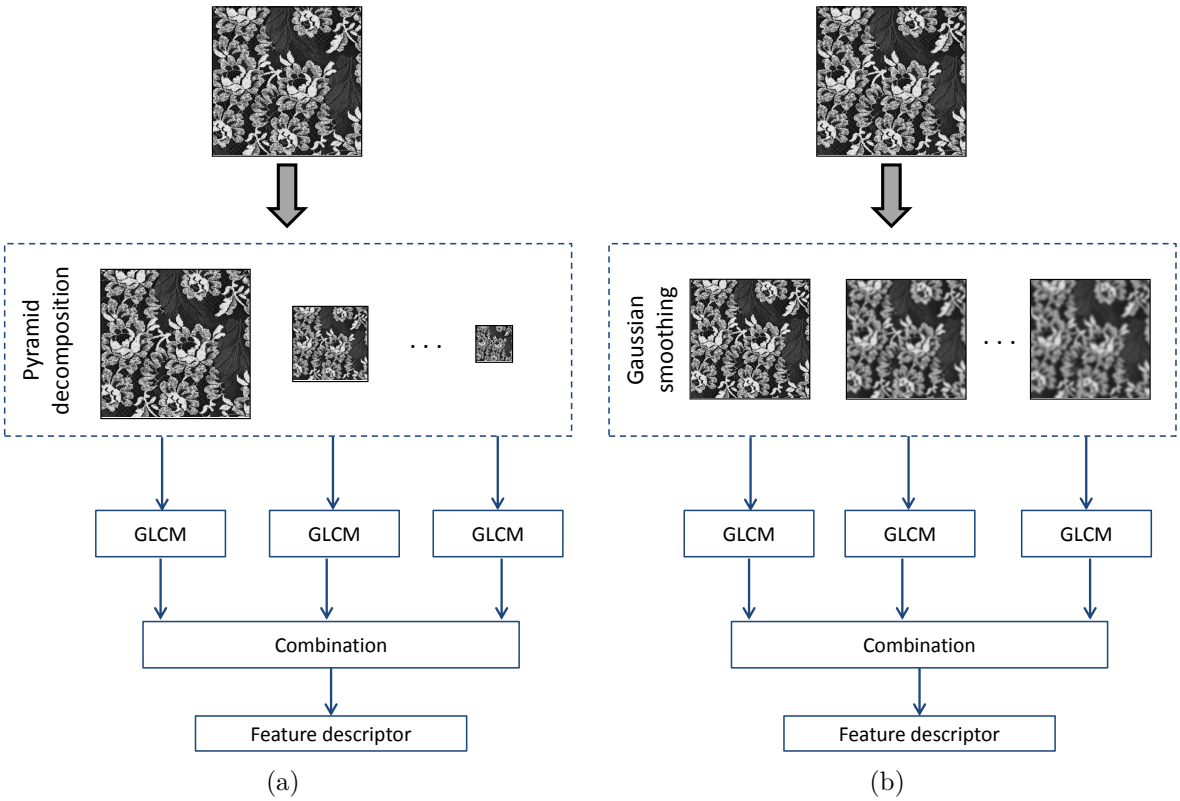


Figure 4.1: Multi-scale approaches for GLCM. (a) Pyramid decomposition; (b) Gaussian smoothing.

all the information.

In the second method, referred to as *local*, a normalization is applied to each feature descriptor extracted from a certain scale, then all normalized features are joined together through a simple concatenation.

The third scheme, *(L + G)*, is achieved by applying a normalization to the feature descriptors extracted from each scale, followed by a global normalization of all features united in a single feature vector.

The fourth strategy, referred to as *corresponding*, normalizes corresponding elements of the scales in the descriptor, that is, the first feature extracted from the first scale is normalized in relation to the first features of every other scale, then the second feature is normalized according to the second feature of each scale, and so forth. This combination is adequate when features have very discrepant value intervals, even in a single scale.

Finally, since all the previous multi-scale combination strategies increase the size of the final feature descriptor, a fifth method, referred to as *weighted*, is proposed to maintain the size of the descriptor equal to the number of features extracted from a single scale. Instead of a normalization, as used in the fourth method, a weighted mean is used to

combine the values of corresponding elements of each scale. Each scale will be assigned with decreasing weight of value $1/2^k$, where k is the number of the current scale.

4.4 Experimental Results

This section describes and discusses the results obtained through the evaluation of the feature descriptors applied to texture classification. All experiments ere conducted on an Intel Core i7-2630QM processor, 2.2 GHz with 8 GB of RAM running 64-bit Windows operating system. The method was implemented using C++ programming.

We consider five texture data sets: UIUC [60], UMD [118], Brodatz [15], OuTex [74] and VisTex [121]. All data sets were used to estimate and evaluate the parameters employed to assess the effectiveness of the proposed multi-scale feature descriptor for texture classification.

Section 4.4.1 presents a brief explanation of each data set used in our experiments and the classification protocols employed. In Section 4.4.2, we describe the parameter values used for each method. Then, evaluations of the proposed approach and comparisons to other feature extraction methods are presented and discussed in Section 4.4.3.

4.4.1 Data Sets

This section describes the main characteristics of the five data sets used in our experiments, such that four of them (UMD, UIUC, VisTex and OuTex) were detailed in Chapter 3 with the addition of the Brodatz data set.

Brodatz Data Set

The Brodatz [15] photo album is a widely used texture data set, often treated as a benchmark for texture classification. It contains 111 different texture classes with 512 × 512 pixels, which are subdivided to compose several classes. The test suite Contrib_TC_00004 is used in this work (available at the OuTex site [78]). It is composed of 2048 images of 64 × 64 pixels divided equally among 32 classes. Ten combinations for training and test are considered for this suite. Figure 4.2 shows a mosaic containing samples of Brodatz data set.

4.4.2 Experimental Setup

For all experiments, the nearest neighbor classifier is applied after all variables are normalized to present zero mean and unit variance to reduce potential predominance among values of distinct features. In addition, Principal Component Analysis (PCA) is applied

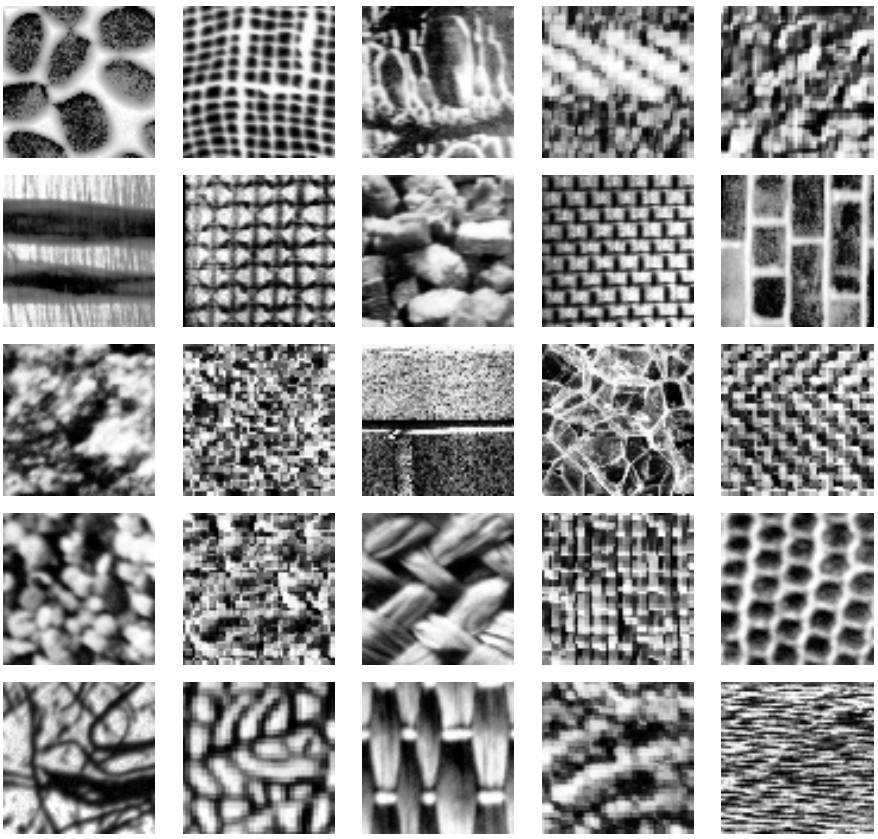


Figure 4.2: Examples of texture samples extracted from Brodatz data set [15].

to avoid dimensionality issues – the estimation of the best dimensionality is performed by cross-validation and it is allowed to vary between 5 and 40 dimensions. The dimensionality which achieved the best classification rate for the data set was determined and chosen, whose values are presented in Table 4.4.

A subset of 12 descriptors from the original 14 descriptors described in Section **??** is considered for the GLCM: angular second moment, contrast, correlation, sum of squares, inverse difference moment, sum average, sum variance, sum entropy, entropy, difference variance, difference entropy and maximal correlation coefficient. Co-occurrence matrices for four orientations are computed (0^o, 45^o, 90^o and 135^o) and used to compose the feature vector with 48 dimensions (12 per orientation).

The values for parameters d and the number of bins for the GLCM are experimentally estimated to be 1 and 256, respectively (these values are used throughout the experiments). Table 4.1 shows the parameter estimation. According to the table, the higher the number of bins, the better are the results for most data sets. However, the displacement parameter d has not presented the same behavior. For data sets with larger samples such as UMD and UIUC, larger values of the displacement provide better results, while

the opposite behavior takes place in data sets with samples with small size (VisTex and OuTex).

Regarding the remaining feature descriptors that will be considered in the comparison, for the granulometry we used kernel size equal to 25 and step equal to 2 in UIUC and UMD data sets, and kernel size equal to 5 and step equal to 1 for the remaining data sets, which leads to feature vectors with 104 and 40 dimensions, respectively. In Gabor filters, we used 8 orientations and 5 scales, generating 40 filters for convolution. For each filtered image, two measures were extracted (mean and standard deviation), which lead to 80 descriptors. The LBP and ILBP were used with their standard configuration and the resulting feature vectors have 256 and 512 dimensions, respectively.

4.4.3 Evaluations and Comparisons

We have evaluated the number of decomposition levels, approaches used to decompose the images into multiple scales, and strategies for combining the feature descriptors extracted from different scales. The results are shown in Tables 4.2 and 4.3. According to them, the scale decomposition based on Gaussian smoothing has provided higher classification rates for all data sets and the most significant difference occurs with the OuTex data set (84.61% with Pyramid decomposition and 89.50% with Gaussian smoothing). This is because the small sample size for this data set associated with the pyramid decomposition results in the computation of the GLCM for very small images, resulting, therefore, on sparse matrices. The Gaussian smoothing decomposition is considered during the comparison to other methods.

Tables 4.2 and 4.3 also evaluate five approaches to combine the descriptors extracted from each scale. According to the results, three approaches have achieved higher classification rates: local normalization, normalization based on corresponding descriptors, and simple concatenation without normalization. The combination strategy, based on reducing the weight of the descriptors extracted from smaller scales, showed a weak performance (meaning that the importance of all scales should be equally considered). The combination strategy that achieved the best results for each data set is used in the comparisons to other methods.

Finally, comparisons among the proposed multi-scale feature descriptor with other well-known methods for texture analysis are shown in Table 4.4. For the UMD and UIUC data sets, we show the results when a different number of samples are used for training (as indicated in the second column). The remaining datasets follow the protocols described in Section 4.4.1.

There are important observations that can be made according to the results shown in Table 4.4. First, significant improvements have been achieved for all data sets when the

Data Set	d	Bins					
		8	16	32	64	128	256
UMD	1	90.07	90.86	91.60	91.62	91.29	91.56
	2	91.49	92.36	92.66	92.44	92.17	92.01
	3	91.90	92.60	92.68	92.52	92.59	**92.78**
UIUC	1	65.15	67.23	68.23	68.21	68.83	71.38
	2	71.44	72.47	71.55	72.26	72.27	73.01
	3	73.70	**75.67**	75.56	75.06	74.16	75.53
Brodatz	1	67.06	70.46	72.27	74.24	75.63	76.28
	2	73.37	76.69	78.30	**78.58**	77.97	77.80
	3	65.30	70.31	72.68	76.13	74.59	73.83
VisTex	1	93.75	93.98	93.52	93.75	93.98	**94.44**
	2	93.98	93.52	92.82	92.82	93.52	93.29
	3	90.74	90.97	90.51	90.97	91.20	91.20
OuTex	1	77.56	78.39	81.18	80.98	82.22	**83.02**
	2	78.56	78.13	79.83	79.92	80.75	82.19
	3	77.08	75.61	77.07	76.86	78.35	80.00

Table 4.1: Parameter estimation for all data sets considering single scale GLCM (20 training samples were considered for UMD and UIUC data sets). The results show classification rates, in percentages.

multi-scale GLCM is employed, compared to the original approach. Second, considering the other methods in the literature, the proposed approach achieved the best classification rates on three out of the five tested data sets. Furthermore, it is also important to point out that the most significant improvements were achieved on data sets presenting samples with large sizes (UMD and UIUC), in which more information can be captured by a multi-scale approach. Finally, although the Gabor filters also perform multi-scale analysis, the classification rates are not as high as the ones achieved by the proposed approach.

Table 4.5 shows the computational time required for each feature in all tested data sets. It can be observed that multi-scale GLCM takes more time than its single-scale version, which is expected since that the GLCM has to be computed in every level of the pyramid in the proposed method.

Data Set	Scales	Combination Approach				
		Local	L + G	Corresponding	Weighted	Concatenation
UMD	2	90.84	88.27	92.81	91.67	93.03
	3	88.26	88.14	**95.28**	91.74	94.07
	4	89.88	89.88	94.65	91.77	94.52
UIUC	2	71.82	68.91	68.35	70.60	75.38
	3	72.18	72.18	71.93	70.99	79.02
	4	73.19	73.99	71.07	72.12	**80.37**
Brodatz	2	**68.27**	**68.27**	67.45	67.80	68.03
	3	65.77	65.77	64.10	66.72	61.81
	4	58.13	58.13	52.08	60.56	53.18
VisTex	2	90.28	90.28	86.11	93.98	**94.68**
	3	90.05	90.05	80.79	93.06	94.91
	4	87.04	87.04	68.98	93.75	92.59
OuTex	2	**84.61**	**84.61**	73.79	83.26	84.29
	3	82.28	82.28	70.94	82.22	82.44
	4	76.66	76.66	67.47	82.01	74.33

Table 4.2: Classification rates (%) achieved when the pyramid decomposition is considered for the GLCM and multiple approaches for feature descriptor combination are employed, as described in Section 4.3 (due to limited space, we used the term L+G, which means local followed by global normalization).

Data Set	Scales	Combination Approach				
		Local	L + G	Corresponding	Weighted	Concatenation
UMD	2	92.40	88.33	90.59	91.99	93.25
	3	88.25	88.25	93.89	92.29	94.28
	4	88.46	88.46	**95.28**	92.42	94.94
UIUC	2	80.08	80.08	81.88	81.74	81.02
	3	80.13	80.13	81.79	81.88	81.85
	4	81.88	81.88	81.74	81.88	**81.79**
Brodatz	2	80.82	80.82	86.37	73.50	81.82
	3	81.21	81.21	87.07	76.45	83.48
	4	81.29	81.29	**87.19**	77.71	83.05
VisTex	2	94.44	91.20	94.21	94.21	**96.06**
	3	91.90	91.90	93.98	93.52	95.14
	4	91.90	91.90	**96.06**	93.06	95.37
OuTex	2	88.71	88.71	84.98	84.56	88.30
	3	**89.50**	**89.50**	87.09	84.94	88.75
	4	89.33	89.31	87.36	84.65	88.38

Table 4.3: Classification rates (%) achieve when the Gaussian smoothing is considered for the GLCM and multiple approaches for feature descriptor combination are employed, as described in Section 4.3 (due to limited space, we used the term L+G, which means local followed by global normalization).

Data Set	Samples	Feature Extraction Method					
		LBP	ILBP	Gabor	Granulometry	GLCM	
						original	multi-scale
UMD	5	72.5	73.3	75.7	75.4	83.1	**85.2**
	10	81.5	81.3	82.3	83.8	88.7	**91.2**
	15	85.4	85.0	84.9	88.0	91.1	**93.5**
	20	88.1	87.4	88.1	90.6	92.7	**95.2**
UIUC	5	41.7	44.8	51.0	48.3	58.9	**65.1**
	10	44.6	51.7	59.4	58.9	68.5	**74.4**
	15	49.2	54.7	65.1	64.5	72.7	**78.9**
	20	53.4	61.3	69.9	68.6	75.6	**81.7**
Brodatz	-	87.1	**89.4**	**89.4**	88.8	78.3	87.2
VisTex	-	96.5	**97.9**	92.6	94.4	94.4	96.0
OuTex	-	89.0	82.0	84.9	89.2	83.1	**89.4**

Table 4.4: Texture classification results achieved by multiple texture descriptors for different data sets. The second column indicates the number of samples used during training for the UMD and UIUC data sets. The results show classification rates in percentage (values vary from the previous analysis due to different parameter variation used in these experiments).

Data Set	Feature Extraction Method					
	LBP	ILBP	Gabor	Granulometry	GLCM	
					original	multi-scale
UMD	94.86	105.85	4568.99	9979.91	135.84	633.43
UIUC	24.95	27.21	1371.59	2536.75	39.35	183.10
Brodatz	0.78	0.88	110.12	68.51	12.30	56.05
VisTex	1.95	2.14	95.17	116.60	8.05	32.16
OuTex	1.89	3.96	297.94	79.20	41.66	180.49

Table 4.5: Computational time (in seconds) required to perform the feature extraction.

4.5 Conclusions

This paper describes an extension of the GLCM texture descriptor to multiple scales based on a Gaussian smoothing approach and a pyramid decomposition. Features extracted from GLCM at different scales are combined and evaluated for each multi-resolution representation.

The performance of the proposed approach is evaluated by applying the multi-scale descriptor on five benchmark texture data sets and the results are compared to other well-known descriptors for texture analysis.

Although the pyramid decomposition showed improvement on the results in some cases, such enhancement was not kept by adding more scales in all datasets. We believe this behavior occurs due to the small sample size. Such behavior is consistent with the results obtained in the high resolution data sets, which showed more significant improvements. On the other hand, Gaussian smoothing presented better results for every added scale, achieving better performance.

The strategy for the scales that achieved the best results was the combination of corresponding elements, probably due to the distinct information obtained with the GLCM descriptor, where every single feature has a very different value. However, no conclusion can be drawn regarding the use of combination in a generic way.

The proposed Gaussian smoothing multi-scale approach to GLCM achieved significant improvements, outperforming single-scale GLCM in all tested data sets.

Chapter 5

Adaptive Detection of Human Skin in Color Images

Preamble

Detection of human skin has several practical applications in image and video processing fields, such as face detection, image indexing, gesture recognition, and nudity detection. The purpose of a human skin detection method is to distinguish image portions between skin and non-skin regions. Many existing approaches in the literature are based on features that explore information of color, shape and texture in the images. Furthermore, several skin detection methods do not produce satisfactory results since they classify image regions based on fixed thresholds and are sensitive to illumination conditions. This paper describes a method for adaptive detection of human skin in images based on a normalized lookup table. The resulting probability map is used to detect skin and non-skin regions, which are refined through texture descriptors. Experimental results show the effectiveness of the proposed method.

5.1 Introduction

Detection of human skin has applications in several areas, such as face recognition, gesture analysis, nudity detection, person tracking, content-based image retrieval, among others. The presence of people in an image or a video scene can be evidenced by finding skin regions.

Automatic skin detection is a challenging task, especially under varying illumination and partial occlusions [88]. Another inherent difficulty is that skin tones can significantly vary across individuals. Several methods found in the literature are based on fixed thresh-

olds [20, 35] and are sensitive to geometric variations of skin patterns and are not robust to image resolution changes.

This paper describes and evaluates an adaptive human skin detection method based on a normalized lookup table, whose resulting probability map is used to detect skin and non-skin regions in the images. Such regions are then refined through texture descriptors. Experiments are conducted to apply the methodology to several images and results show the effectiveness of the proposed method.

The text is organized as follows. Section 5.2 presents the main concepts and work related to skin detection. Section 5.3 describes the proposed method for adaptive detection of human skin. Experimental results are shown in Section 5.4. Finally, Section 6 concludes the paper with final remarks.

5.2 Background

Several methods for skin detection have been proposed in the literature [55]. They can be categorized as pixel-based and region-based approaches. Pixel-based methods classify each pixel as skin or non-skin without considering its neighborhood, whereas region-based methods explore the spatial organization of neighbor skin pixels to improve the skin detection process. Comprehensive surveys on skin color modeling and skin detection can be found in the literature [55, 93, 120].

A skin color model is commonly used to identify if a pixel or region is skin or non-skin. Various color spaces have been used in skin detection, such as RGB [53, 126], HSV [12], CIE Lab [19], CIE Luv [128], YES [100] and YCbCr [20, 123].

Zarit et al. [130] compared five color spaces and two non-parametric methods for skin modeling. Shin et al. [105] examined eight color spaces applied to skin detection. Albiol [3] presented a theoretical proof that there is an optimum skin detector for every color space such that all skin detectors have the same performance. Brand and Mason [14] evaluated three skin color modeling strategies. Lee and Yoo [61] compared two most popular parametric skin models in different chrominance spaces.

Brown et al. [16] proposed a statistical model for skin color distribution based on a Self-Organizing Map (SOM). Sigal et al. [106] used a Markov model with adaptive color histogram to calculate the skin color distribution. Yang and Ahuja [128] described the use of expectation-maximization (EM) technique applied to Gaussian mixture models (GMM) for skin detection.

In contrast with methods described previously, the proposed method detects skin regions in the images through three consecutive stages, where each step is improved with the previous one. Measures of color, homogeneity and texture are taken into the account along the skin detection process.

5.3 Methodology

In this section, we present the process proposed to detect skin in images. The proposed approach is based not only of color information as traditional skin detection methods, but also on homogeneity and texture information to perform a more accurate detection.

First, we create a normalized lookup table (LUT) [53]. This is done by collecting measures of skin and non-skin pixel color samples and arranging them in a normalized histogram. This histogram provides a probability indicating how likely each pixel is skin or non-skin, such that a probability map is created to the entire image. With the application of a proper threshold, this map can be used to detect whether each pixel is skin or not.

To make the detection process more adaptable and achieve better results, a measure of how homogeneous is the detected region is evaluated since human skin regions tend to be more homogeneous than other types of surfaces [90]. Thus, the achieved results are maintained in an assessment of how homogeneous every region is according to the probability map, that is, regions that are not considered homogeneous are discarded, whereas homogeneous regions are grown as long as they remain homogeneous and then used in the output.

Finally, after considering two skin properties, color and homogeneity, the detection process is refined by taking texture information into account. Figure 5.1 shows the main stages of the proposed method.

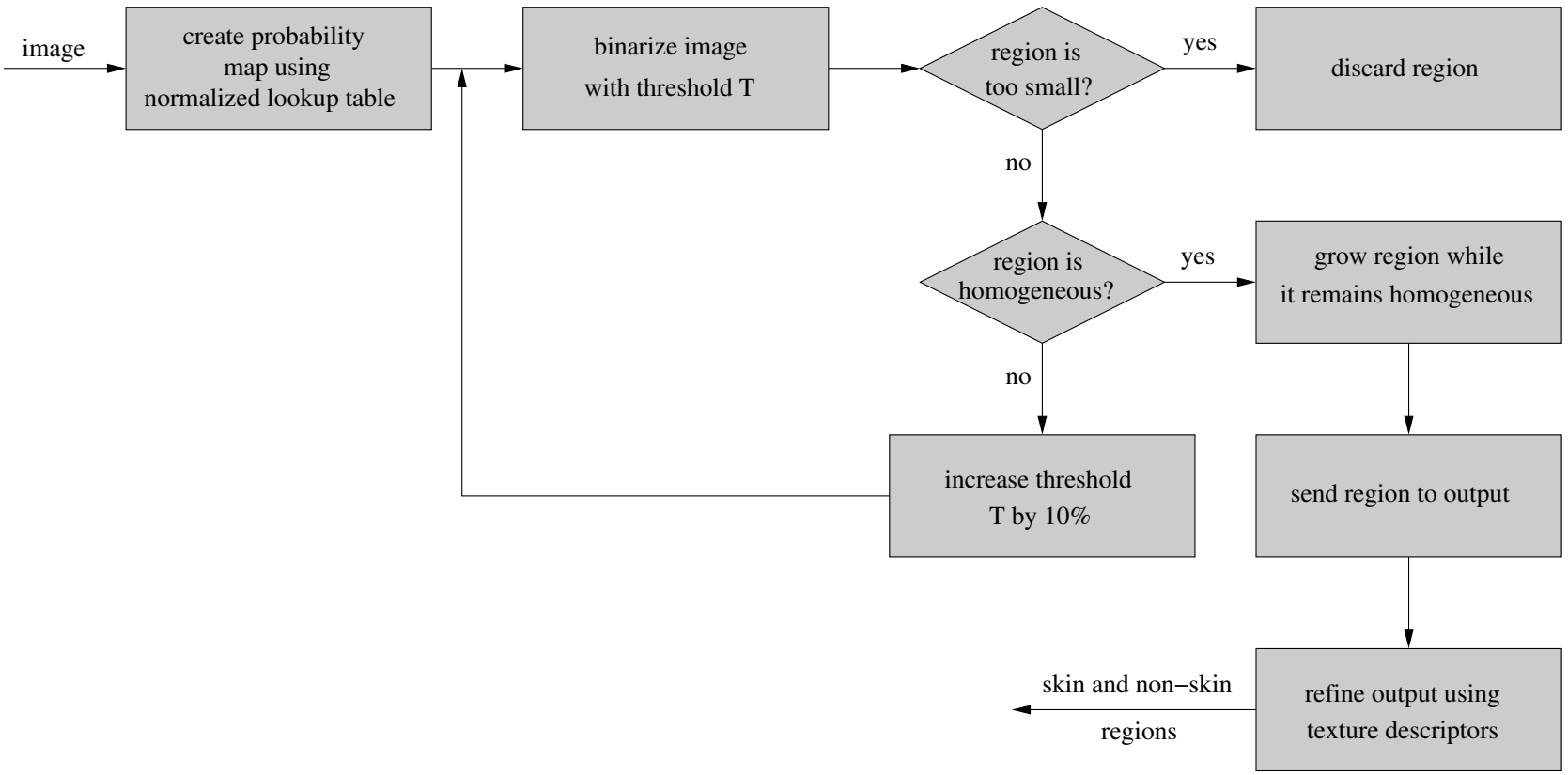


Figure 5.1: Diagram with the main stages of the skin detection process.

5.3.1 Construction of Human Skin Color Model

To formulate the skin color model, we used patches of several images collected from the Internet. These patches were manually selected in skin and non-skin groups. Several non-skin samples were extracted from the Caltech image dataset [58].

Color histograms were constructed through both the skin and non-skin groups of images. RGB images were used to construct these histograms, each pixel forms a vector [RGB] which is translated to the lookup table as

$$H([RGB]) = H(R + [G * 256] + [B * 256 * 256]) \tag{5.1}$$

To find a probability of a pixel being in each group

$$P([RGB]) = \frac{H([RGB])}{\sum_{n=1}^{|H|} H(n)} \tag{5.2}$$

For every test image, we use a threshold t and the LUT to classify the image as follows [33]

$$\text{if } \frac{P(\text{pixel} \mid \text{skin})}{P(\text{pixel} \mid \neg \text{ skin})} \geq t \text{ then } P \text{ is labeled as skin} \tag{5.3}$$

where $P(\text{pixel} \mid \text{skin})$ is the probability of a pixel containing skin (informed by the histogram of the skin group) and $P(\text{pixel} \mid \neg \text{ skin})$ is the probability of a pixel being in the non-skin group.

5.3.2 Selection of Homogeneous Regions

The LUT produces as an output a color map with the likelihood of every pixel being skin or not skin [90]. At this point, we select an initial threshold such that, if the probability of a point in the map is greater than the threshold, then it is labeled as skin, otherwise it is labeled as non-skin. Then, we create a black and white representation of the image (black being non-skin and white being skin). This gives us several skin-labeled connected components which will form a region. For each region of the image, we apply the following steps:

- if the region is smaller than a given threshold, it is discarded (the threshold size varies with the size of the image).

- while the region is not homogeneous, the threshold is increased by 10%.

- when a homogeneous region is found, we then dilate this selected image portion using a 3×3 square structure element as long as it remains homogeneous and is still smaller than the bounding box of the region.

- this homogeneous region is added to the skin list.

A region of the image is assigned as homogeneous if it has the following properties [90]:

$$(\sigma < S_t) \text{ AND } ((\frac{N_e}{N_d} \leq N_{dT}) \text{ OR } (\frac{N_e}{N_s} \leq N_{sT})) \tag{5.4}$$

where σ is the standard deviation of the pixel colors in the region, N_e is the number of edge pixels (found by using Sobel detector [38] in the entire input image) inside the region (boundary pixels are excluded), N_s the is number of skin pixels in the region and N_d is the maximum dimension of the region bounding box (either width or height, whichever is the largest). S_t, N_{dT} and N_{sT} are thresholds empirically chosen in the experiments, as detailed in Section 5.4.

5.3.3 Refinement with Texture Descriptors

After the previously described process has been applied to the images, we use a classifier trained to detect skin textures. The training was conducted on a Quadratic Discriminant Analysis (QDA) classifier [62] with the same images used to create the LUT histograms.

To perform the tests, we used a small window (32×32) sliding through the image, assigning how likely each window is of containing skin, creating a skin likelihood map. Then, we used a varying threshold to check the accuracy of the classification results.

5.4 Experimental Results

The experiments were performed by applying the proposed methodology described in Section 5.3. For the training of both normalized lookup table (LUT) [53] and quadratic discriminant analysis (QDA) classifier [62], we used randomly selected samples from the Internet and images extracted from the Caltech image dataset [58].

To form the homogeneity measure, the following variations of the parameters were employed: 35, 40 and 45 for S_t , 1.5, 2.5 and 3.5 for N_{dt} and 0.02, 0.12 and 0.22 for N_{st}. For the Sobel filter, we used the following thresholds: 20, 90, 160 and 230. We discarded any region that is smaller than a threshold in a range of 250, 500, 750 and 1000, varying with the size of the image (larger images used higher values).

The tests were conducted on the MCG skin-dataset [49], which contains 1000 images containing humans with skin exposure. The authors provide a ground truth for evaluation

purposes. Each image was applied to the proposed method, producing an output that was compared against the ground truth. True positive and false positive rates were computed, such that the best combinations for each of the parameters were selected in a small subgroup of the original dataset, then applied to the full one. The values are shown in Table 5.1.

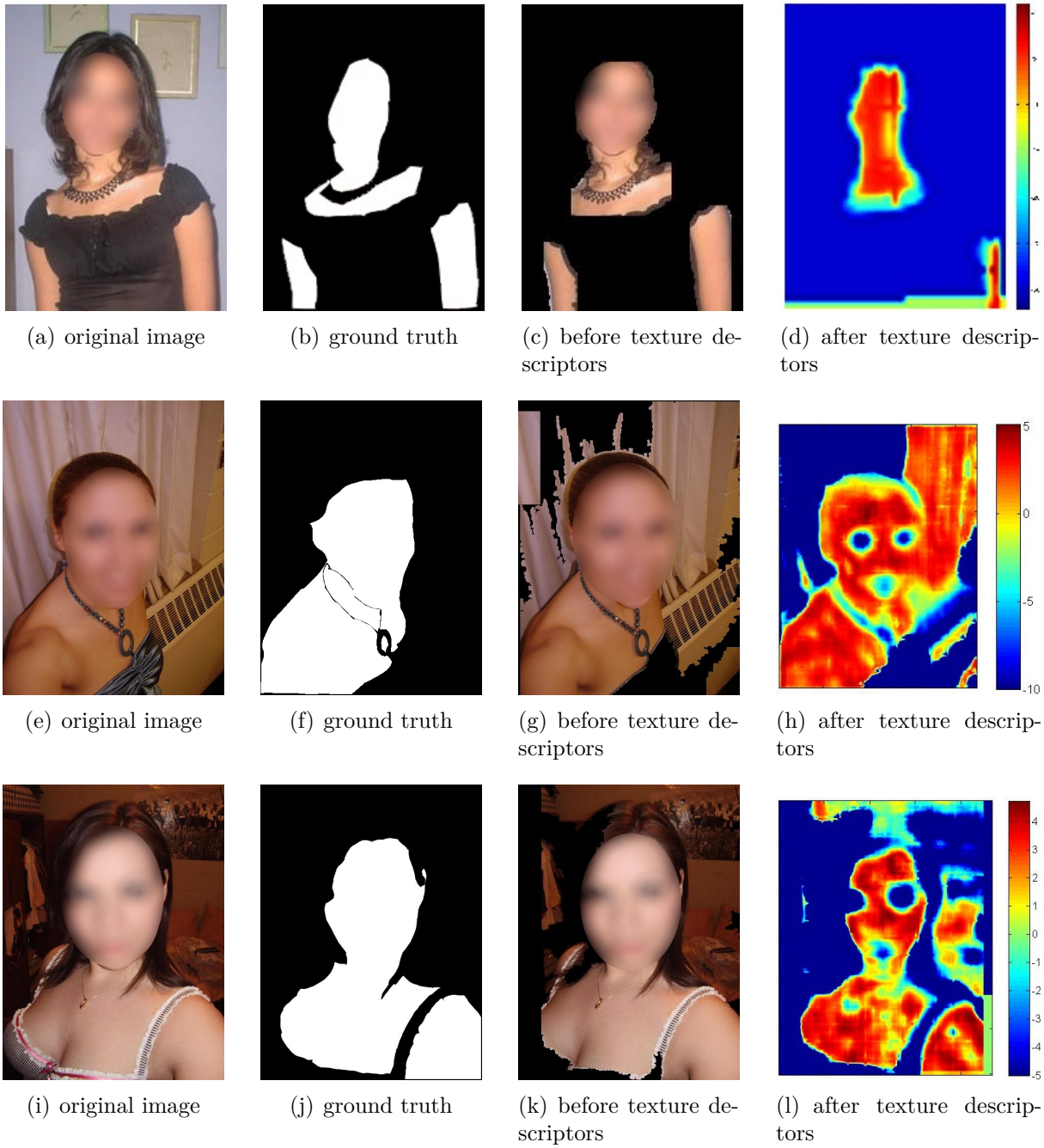
Textures	1	2	3	4	5	6
N_{dt}	1.5	2.5	3.5	1.5	2.5	3.5
N_{st}	0.22	0.22	0.22	0.02	0.02	0.02
Sobel threshold	20	20	20	90	90	90
S_t	40	40	45	45	45	40
True Positive Rate	67.14%	73.86%	77.81%	84.59%	85.50%	88.22%
False Positive Rate	13.08%	15.10%	16.78%	19.37%	20.62%	23.18%

Table 5.1: Values of the most distinguishable parameter combinations.

Figure 5.2 illustrates the results by applying the proposed method to three different image samples. In the first row, it is possible to see that the result after applying the texture descriptors was not improved in relation the result obtained with skin color model and selection of homogeneous regions. On the other hand, in the second and third rows, the texture descriptors improved the result obtained in the previous stage, which increases the accuracy of the classifier.

Finally, to conduct the experiments with the texture classifier, the following parameters were used: $S_t = 40$, Sobel threshold = 90, $N_{st} = 0.02$ and $N_{dt} = 3.5$. This parameter combination produced the best true positive values, subjected to further refinement.

We assessed three different texture descriptors: Local Binary Patterns (LBP) [76], Grey Level Co-occurrence Matrices (GLCM) [41] and a Multiscale GLCM [108]. A threshold was used to build the likelihood map into skin and non-skin regions. This threshold was sampled between -30 and 7 (minimum and maximum likelihood, respectively), each sample providing a binary image compared to the ground truth to produce true and false positive measures, where several points were collected from each measure. Figure 5.3 shows a comparison among these textures detections along with the results obtained by not using any texture detection. According to the results, while the LBP feature descriptor achieved the lowest results, the remaining approaches showed similar performances.



(a) original image (b) ground truth (c) before texture descriptors (d) after texture descriptors

(e) original image (f) ground truth (g) before texture descriptors (h) after texture descriptors

(i) original image (j) ground truth (k) before texture descriptors (l) after texture descriptors

Figure 5.2: Results for two samples of images. (a), (e) and (i) original images; (b), (f) and (j) ground truth of the images; (c), (g) and (k) results after applying skin color model and selecting homogeneous regions; (d), (h) and (l) probability maps after applying texture descriptors.

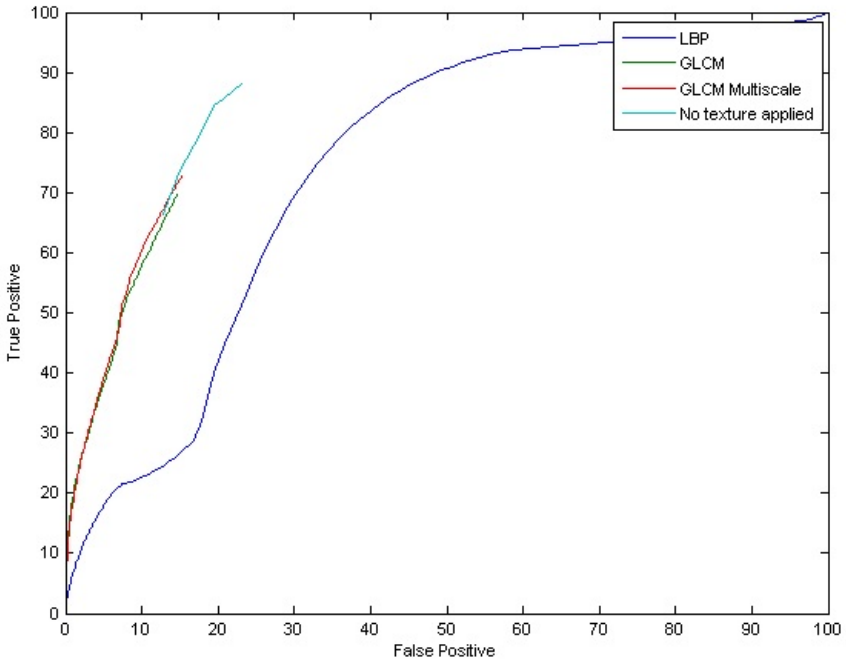


Figure 5.3: Results after applying different texture descriptors.

5.5 Conclusions and Future Work

This paper presented an adaptive human skin detection method based on a probability map used to detect skin and non-skin regions.

It was experimentally found that the texture stage has the potential to make some improvements on the results. However, in images that already presented high detection rate after the detection of homogeneous regions, the texture stage caused negative effect. The reason for that is probably due to the training process that was not sufficient to promote accurate distinction between region and non-regions. The training with a broader spectrum of skin images can improve the results.

As future directions, we intend to increase the number of training texture samples as an attempt to decrease the negative effect on some images and also improve the proposed method through a face detection process to adaptively enhance the skin color model to each image under different illumination conditions.

Chapter 6

Conclusions and Future Work

Several applications in image and video processing, such as image segmentation, image classification, content-based image retrieval, scene recognition, video analysis, among others, are based on the use of image descriptors, in particular, texture descriptors. Despite the large number of texture descriptors proposed in the literature, the definition of texture in images remains a challenging task.

The main objective of this work was to propose an enhancement to texture descriptors based on multi-scale approach. To address such task, we first conducted an investigation on texture descriptors. A detailed analysis was conducted on several descriptors, both at single and multiple scales, ranging various approaches: statistical, signal processing, geometrical and parametric models.

Experimental results demonstrated that ILBP, LBP, Gabor filters, GLCM and granulometry texture descriptors consistently presented high classification rates under various data sets. Moreover, the combination of features, formed by the concatenation of all the tested features together, produced superior results in terms of classification rate when compared to the features individually.

Furthermore, an extension to multi-scale was proposed for a well known and widely used texture descriptor, the GLCM, by considering two variants, the Gaussian smoothing and the pyramid decomposition. The first one is based on applying several layers of Gaussian smoothing and combining them in each image, whereas the latter is based on re-scaling the image size and extracting information from every size and combining the results through the use of several methods: concatenation, local, local + global, normalization of corresponding features, and a weighted mean.

The performance of the proposed approach, as well as all its variants, was evaluated by applying the descriptor to five benchmark texture data sets. The results were compared with the best texture descriptors described previously. The proposed multi-scale pyramid decomposition approach did not prove to be effective with the addition of more scales,

apparently due to the small size of the new scales. For datasets with high resolution images, the gain was superior when every scale was added. On the other hand, the Gaussian smoothing technique proved to boost the results with the increment of new scales, providing significant improvements on all tested data sets over the single-scale version.

For all the tested combinations, the normalization of corresponding features produced superior results, probably due to the distinctness of the information contained in the GLCM feature vector, where every single feature has a very different value than its counterparts.

Finally, the proposed extension to multiple scales was employed as a step in an adaptive human skin detection technique based on a color probability map in order to verify its effectiveness. It was experimentally demonstrated that this stage can provide improvements on several images, however, it failed to provide significant distinction, probably due to the nature of the images employed in the skin detection process, which are pictures taken from a certain distance and usually with not such a good quality in details.

As future directions, we intend to investigate a different approach to incorporating multiple scales to GLCM by varying the distance used to generate the matrix in a single image and also test the multi-scale approach on a remote sensing context to verify its adaptability. We also plan to create a new set of images that is inherently composed of multiple scales for the same texture object to confirm the adaptability of these features in a multiple scale environment. Finally, the skin detection method could be improved by training more samples of skin and non-skin, as well as by using face detection to adaptively enhance the skin color model according to the image, which will provide a better detection in images under varying illumination conditions.

Bibliography

[1] J. Ahlberg. A system for face localization and facial feature extraction. Technical report, Dept. of Electrical Engineering, Linköping University, 1999.

[2] O. Alata, C. Cariou, C. Ramananjarasoa, and M. Najim. Classification of rotated and scaled textures using HMHV spectrum estimation and the Fourier-Mellin transform. In *International Conference on Image Processing*, volume 1, pages 53–56, 1998.

[3] A. Albiol, L. Torres, and E. Delp. Optimum color spaces for skin detection. In *International Conference on Image Processing*, volume 1, pages 122–124, Thessaloniki, Greece, Oct. 2001.

[4] F. Andrade, J. Almeida, H. Pedrini, and R. Torres. Fusion of Local and Global Descriptors for Content-Based Image and Video Retrieval. In *17th Iberoamerican Congress on Pattern Recognition*, volume 7441, pages 845–853, Buenos Aires, Argentina, 2012.

[5] V. Arvis, C. Debain, M. Berducat, and A. Benassi. Generalization of the Cooccurrence Matrix for Colour Images: Application to Colour Texture Classification. *Image Analysis & Stereology*, 23:63–72, 2004.

[6] J. Babaud, A. P. Witkin, M. Baudin, and R. O. Duda. Uniqueness of the gaussian kernel for scale-space filtering. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 8:26–33, January 1986.

[7] A. Baraldi and F. Parmiggiani. An Investigation of the Textural Characteristics Associated with Gray Level Cooccurrence Matrix Statistical Parameters. *IEEE Transactions on Geoscience and Remote Sensing*, 33(2):293–304, Mar. 1995.

[8] H. Bay, T. Tuytelaars, and L. Van Gool. SURF: Speeded-Up Robust Features. In *European Conference on Computer Vision*, pages 346–359, 2006.

[9] S. Belongie, J. Malik, and J. Puzicha. Shape Matching and Object Recognition Using Shape Contexts. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24:509–522, 2002.

[10] M. Benco and R. Hudec. Novel Method for Color Textures Features Extraction based on GLCM. *Radioengineering*, 4(16):64–67, 2007.

[11] J. E. Besag. Spatial Interaction and the Statistical Analysis of Lattice Systems. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 36(2):192–236, May 1974.

[12] N. Bojic and K. Pang. Adaptive skin segmentation for head and shoulder video sequence. In *SPIE Visual Communications and Image Processing*, Perth, Australia, June 2000.

[13] A. Bosch, A. Zisserman, and X. Munoz. Representing Shape with a Spatial Pyramid Kernel. In *6th ACM International Conference on Image and Video Retrieval*, pages 401–408, Amsterdam, The Netherlands, 2007.

[14] J. Brand and J. Mason. A Comparative Assessment of Three Approaches to Pixel-level Human Skin-Detection. In *International Conference on Pattern Recognition*, volume 1, pages 1056–1059, Barcelona, Spain, Sept. 2000.

[15] P. Brodatz. *Textures: A Photographic Album for Artists and Designers*. Dover Publications, Inc., New York, USA, 1966.

[16] D. Brown, I. Craw, and J. Lewthwaite. A SOM Based Approach to Skin Detection with Application in Real Time Systems. In *British Machine Vision Conference*, Manchester, United Kingdom, Sept. 2001.

[17] P. J. Burt. Fast filter transform for image processing. *Computer Graphics and Image Processing*, 16(1):20 – 51, 1981.

[18] P. J. Burt, Edward, and E. H. Adelson. The laplacian pyramid as a compact image code. *IEEE Transactions on Communications*, 31:532–540, 1983.

[19] J. Cai, A. Goshtasby, and C. Yu. Detecting human faces in color images. In *IEEE Internatinal Workshop on Multimedia Database Management Systems*, pages 124–131, Austin, TX, USA, Aug. 1998.

[20] D. Chai and K. Ngan. Face segmentation using skin-color map in videophone applications. *IEEE Transactions on Circuits and Systems for Video Technology*, 9(4):551–564, June 1999.

[21] D. Clausi and M. Jernigan. A Fast Method to Determine Co-occurrence Texture Features. *IEEE Transactions on Geoscience and Remote Sensing*, 36(1):298–300, 1998.

[22] G. R. Cross and A. K. Jain. Markov Random Field Texture Models. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 5(1):25–39, Jan. 1983.

[23] J. Crowley. A representation for visual information. Technical Report CMU-RI-TR-82-07, Robotics Institute, Pittsburgh, PA, November 1981.

[24] N. Dalai, B. Triggs, I. Rhone-Alps, and F. Montbonnot. Histograms of Oriented Gradients for Human Detection. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 886–893, 2005.

[25] I. Daubechies. Orthonormal bases of compactly supported wavelets. *Communications on Pure and Applied Mathematics*, 41(7):909–996, 1988.

[26] I. Daubechies. *Ten Lectures on Wavelets*. CBMS-NSF Regional Conference Series in Applied Mathematics. Society for Industrial and Applied Mathematics, Philadelphia, PA, USA, 1992.

[27] J. G. Daugman. 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*, 2:1160–1169, 1985.

[28] E. R. Davies. Introduction to Texture Analysis. In J. S. Majid Mirmehdi, Xianghua Xie, editor, *Handbook of Texture Analysis*, pages 1–31. Imperial College Press, 2008.

[29] M. Do and M. Vetterli. The Contourlet Transform: An Efficient Directional Multiresolution Image Representation. *IEEE Transactions on Image Processing*, 14(12):2091–2106, 2005.

[30] J. M. H. du Buf, M. Kardan, and M. Spann. Texture feature performance for image segmentation. *Pattern Recogn.*, 23(3-4):291–309, Mar. 1990.

[31] A. Elgammal, C. Muang, and D. Hu. Skin Detection: A Short Tutorial. *Encyclopedia of Biometrics*, 2009.

[32] W. T. Freeman and E. H. Adelson. The Design and Use of Steerable Filters. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 13:891–906, 1991.

[33] K. Fukunaga. *Introduction to Statistical Pattern Recognition*. Computer Science and Scientific Computing. Elsevier Science, 1990.

[34] M. M. Galloway. Texture Analysis Using Gray Level Run Lengths. *Computer Graphics and Image Processing*, 4:172–179, 1975.

[35] C. Garcia and G. Tziritas. Face detection using quantized skin color regions merging and wavelet packet analysis. *IEEE Transactions on Multimedia*, 1(3):264–277, Sept. 1999.

[36] A. Gelzinis, A. Verikas, and M. Bacauskiene. Increasing the Discrimination Power of the Co-occurrence Matrix-based Features. *Pattern Recognition*, 40(9):2367–2372, 2007.

[37] S. Geman and D. Geman. Stochastic Relaxation, Gibbs Distributions, and the Bayesian Restoration of Images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 6(6):721–741, Nov. 1984.

[38] R. Gonzalez, R. Woods, and S. L. Eddins. *Digital Image Processing using MATLAB*. Gatesmark Publishing, 2009.

[39] A. Hanbury, U. Kandaswamy, and D. A. Adjeroh. Illumination-Invariant Morphological Texture Classification. In C. R. et al., editor, *Mathematical Morphology: 40 Years On*, pages 377–386. Springer, The Netherlands, 2005.

[40] R. M. Haralick. Statistical and Structural Approaches to Texture. *Proceedings of the IEEE*, 67(5):786–804, May 1979.

[41] R. M. Haralick, K. Shanmugam, and I. Dinstein. Textural Features for Image Classification. *IEEE Transactions on Systems, Man and Cybernetics*, 3(6):610–621, Nov. 1973.

[42] D. C. He and L. Wang. Texture Unit, Texture Spectrum, and Texture Analysis. *IEEE Transactions on Geoscience and Remote Sensing*, 28(4):509–512, July 1990.

[43] D. C. He and L. Wang. Texture Features Based on Texture Spectrum. *Pattern Recognition*, 24(5):391–399, May 1991.

[44] H. J. A. M. Heijmans. *Morphological Image Operators*. Academic Press, New York, NY, USA, 1994.

[45] M.-H. Horng, Y.-N. Sunb, and X.-Z. Lin. Texture Feature Coding Method for Classification of Liver Sonography. *Computerized Medical Imaging and Graphics*, 26(1):33–42, Jan. 2002.

[46] R.-L. Hsu, M. Abdel-Mottaleb, and A. Jain. Face detection in color images. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 24(5):696–706, 2002.

[47] Y. Hu. Unsupervised Texture Classification by Combining Multi-Scale Features and K-Means Classifier. In *Chinese Conference on Pattern Recognition*, pages 1–5, 2009.

[48] J. Huang, S. R. Kumar, M. Mitra, W.-J. Zhu, and R. Zabih. Image Indexing Using Color Correlograms. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 762–, 1997.

[49] L. Huang, T. Xia, Y. Zhang, and S. Lin. Human skin detection in images by MSER analysis. In *18th IEEE International Conference on Image Processing*, pages 1257–1260, Brussels, Belgium, 2011.

[50] E. Ising. Beitrag zur Theorie des Ferromagnetismus. *Zeitschrift für Physik A Hadrons and Nuclei*, 31(1):253–258, 1925.

[51] A. K. Jain and F. Farrokhnia. Unsupervised Texture Segmentation Using Gabor Filters. *Pattern Recognition*, 24:1167–1186, 1991.

[52] H. Jin, Q. Liu, H. Lu, and X. Tong. Face Detection Using Improved LBP under Bayesian Framework. In *Third International Conference on Image and Graphics*, pages 306–309, 2004.

[53] M. J. Jones and J. M. Rehg. Statitical Color Models with Application to Skin Detection. *International Journal of Computer Vision*, 46(1):81–96, 2002.

[54] L. Jordao, M. Perrone, J. Costeira, and J. Santos-Victor. Active face and feature tracking. In *International Conference on Image Analysis and Processing*, pages 572–576, 1999.

[55] P. Kakumanu, S. Makrogiannis, and N. Bourbakis. A survey of skin-color modeling and detection methods. *Pattern Recognition*, 40(3):1106–1122, 2007.

[56] N. Kozievitch, J. Almeida, R. Torres, N. Leite, M. Gonçalves, U. Murthy, and E. Fox. Towards a Formal Theory for Complex Objects and Content-Based Image Retrieval. *Journal of Information and Data Management*, 2:321–336, 2011.

[57] E. Kurmyshev and M. Cervantes. A Quasi-Statistical Approach to Digital Image Representation. *Revista Mexicana de Física*, 42(1):104–116, 1996.

[58] R. F. L. Fei-Fei and P. Perona. Learning generative visual models from few training examples: an incremental Bayesian approach tested on 101 object categories. *Workshop on Generative-Model Based Vision - IEEE Conference on Computer Vision and Pattern Recognition*, June 2004.

[59] N.-E. Lasmar, Y. Stitou, and Y. Berthoumieu. Multiscale Skewed Heavy Tailed Model for Texture Analysis. In *16th IEEE International Conference on Image Processing*, pages 2281 –2284, Nov. 2009.

[60] S. Lazebnik, C. Schmid, and J. Ponce. A Sparse Texture Representation Using Local Affine Regions. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 8:1265–1278, 2005.

[61] J. Lee and S. Yoo. An elliptical boundary model for skin color detection. In *International Conference on Imaging Science, Systems and Technology*, Las Vegas, NV, USA, June 2002.

[62] E. L. Lehmann and J. P. Romano. *Testing Statistical Hypotheses*. Springer Texts in Statistics. Springer, New York, third edition, 2005.

[63] M. Leung and A. Peterson. Scale and rotation invariant texture classification. In *Signals, Systems and Computers, 1992. 1992 Conference Record of The Twenty-Sixth Asilomar Conference on*, pages 461–465 vol.1, 1992.

[64] T. Lindeberg. *Scale-Space Theory in Computer Vision*. Kluwer Academic Publishers, New York, NY, USA, 1994.

[65] D. G. Lowe. Distinctive Image Features from Scale-Invariant Keypoints. *International Journal of Computer Vision*, 60:91–110, 2004.

[66] S. G. Mallat. A Theory for Multiresolution Signal Decomposition: The Wavelet Representation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 11(7):674–693, July 1989.

[67] V. Manian and R. Vasquez. Scaled and Rotated Texture Classification using a Class of Basis Functions. *Pattern Recognition*, 31(12):1937 – 1948, 1998.

[68] J. Mao and A. K. Jain. Texture Classification and Segmentation using Multiresolution Simultaneous Autoregressive Models. *Pattern Recognition*, 25:173–188, 1992.

[69] D. J. Marceau, P. J. Howarth, J.-M. M. Dubois, and D. J. Gratton. Evaluation of the grey-level co-occurrence matrix method for land-cover classification using SPOT imagery. *IEEE Transactions on Geoscience and Remote Sensing*, pages 513–519, Mar. 1990.

[70] B. Menser and M. Wien. Segmentation and tracking of facial regions in color image sequences. In *Proc. SPIE*, volume 4067, pages 731–740, 2000.

[71] K. Mikolajczyk and C. Schmid. A Performance Evaluation of Local Descriptors. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27(10):1615 –1630, 2005.

[72] H. Nguyen-Duc, T. Do-Hong, T. Le-Tien, and C. Bui-Thu. A New Descriptor for Image retrieval using Contourlet Co-Occurrence. In *Third International Conference on Communications and Electronics*, pages 169–174, 2010.

[73] P. P. Ohanian and R. C. Dubes. Performance evaluation for four classes of textural features. *Pattern Recognition*, 25(8):819–833, 1992.

[74] T. Ojala, T. Mäenpää, M. Pietikäinen, J. Viertola, J. Kyllönen, and S. Huovinen. Outex - New Framework for Empirical Evaluation of Texture Analysis Algorithms. In *16th International Conference on Pattern Recognition*, volume 1, Washington, DC, USA, 2002.

[75] T. Ojala, M. Pietikainen, and D. Harwood. Performance evaluation of texture measures with classification based on Kullback discrimination of distributions. In *12th IAPR International Conference on Pattern Recognition*, volume 1, pages 582–585, oct 1994.

[76] T. Ojala, M. Pietikainen, and D. Harwood. A Comparative Study of Texture Measures with Classification Based on Featured Distributions. *Pattern Recognition*, 29(1):51–59, Jan. 1996.

[77] T. Ojala, M. Pietikäinen, and T. Mäenpää. Multiresolution Gray-Scale and Rotation Invariant Texture Classification with Local Binary Patterns. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24:971–987, 2002.

[78] OuTex. Dataset, 2011. `http://www.outex.oulu.fi/index.php?page=classification#Outex_TC_00005`.

[79] F. Pacifici and M. Chini. Urban Land-Use Multi-Scale Textural Analysis. In *IEEE International Geoscience and Remote Sensing Symposium*, pages 342–345, 2008.

[80] Y. Pang, W. Li, Y. Yuan, and J. Pan. Fully Affine Invariant SURF for Image Matching. *Neurocomputing*, 85:6–10, May 2012.

[81] Y. Pang, X. Li, and Y. Yuan. Robust Tensor Analysis with L1-Norm. *IEEE Transactions on Circuits and Systems for Video Technology*, 20(2):172–178, 2010.

[82] Y. Pang, X. Li, Y. Yuan, D. Tao, and J. Pan. Fast Haar Transform Based Feature Extraction for Face Representation and Recognition. *IEEE Transactions on Information Forensics and Security*, 4(3):441–450, Sept. 2009.

[83] Y. Pang, H. Yan, Y. Yuan, and K. Wang. Robust CoHOG Feature Extraction in Human Centered Image/Video Management System. *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics*, 42(2):458–468, 2012.

[84] Y. Pang, Y. Yuan, and X. Li. Iterative Subspace Analysis Based on Feature Line Distance. *IEEE Transactions on Image Processing*, 18(4):903–907, Apr. 2009.

[85] G. Pass, R. Zabih, and J. Miller. Comparing Images Using Color Coherence Vectors. In *ACM International Conference on Multimedia*, pages 65–73, 1996.

[86] H. Pedrini and W. Schwartz. *Análise de Imagens Digitais: Princípios, Algoritmos e Aplicações*. Editora Thomson Learning, São Paulo-SP, 2007.

[87] O. Penatti, E. Valle, and R. Torres. Comparative Study of Global Color and Texture Descriptors for Web Image Retrieval. *Journal of Visual Communication and Image Representation*, 23:359–380, 2012.

[88] S. Phung, A. Bouzerdoum, and D. Chai. Skin segmentation using color pixel classification: analysis and comparison. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27(1):148–154, 2005.

[89] S. L. Phung, A. Bouzerdoum, and D. Chai. A novel skin color model in YCbCr color space and its application to human face detection. In *International Conference on Image Processing*, volume 1, pages I–289–I–292 vol.1, 2002.

[90] S. L. Phung, D. Chai, and A. Bouzerdoum. Adaptive skin segmentation in color images. In *IEEE International Conference on Acoustics, Speech, and Signal Processing*, volume 3, pages III–353–6, Hong Kong, China, Apr. 2003.

[91] N. Pican, E. Trucco, M. Ross, D. Lane, Y. Petillot, and I. T. Ruiz. Texture analysis for seabed classification: Co-occurrence matrices vs self-organizing maps. *IEEE Transactions on Computers*, pages 424–428, June 1998.

[92] R. Pingdom. Internet Numbers, 2012. `http://royal.pingdom.com/2013/01/16/internet-2012-in-numbers/`.

[93] C. Prema and D. Manimegalai. Survey on Skin Tone Detection using Color Spaces. *International Journal of Applied Information Systems*, 2(2):18–26, May 2012.

[94] X. Qian, X.-S. Hua, P. Chen, and L. Ke. PLBP: An Effective Local Binary Patterns Texture Descriptor with Pyramid Representation. *Pattern Recognition*, 44:2502–2515, 2011.

[95] X. Qian, G. Liu, D. Guo, Z. Li, Z. Wang, and H. Wang. Object Categorization Using Hierarchical Wavelet Packet Texture Descriptors. In *11th IEEE International Symposium on Multimedia*, pages 44–51, Dec. 2009.

[96] R. S. Torres and A. X. Falcão. Content-Based Image Retrieval: Theory and Applications. *Revista de Informática Teórica e Aplicada*, 13:161–185, 2006.

[97] Y. Rakwatin, P. and Longepe, N. and Isoguchi, O. and Shimada, M. and Uryu. Mapping Tropical Forest using ALOS PALSAR 50m Resolution Data with Multiscale GLCM Analysis. In *IEEE International Geoscience and Remote Sensing Symposium*, pages 1234–1237, 2010.

[98] T. Randen and J. H. Husoy. Filtering for Texture Classification: A Comparative Study. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 21(4):291–310, Apr. 1999.

[99] T. Reed and J. M. H. Dubuf. A Review of Recent Texture Segmentation and Feature Extraction Techniques. *CVGIP: Image Understanding*, 57(3):359–372, May 1993.

[100] E. Saber and A. Tekalp. Frontal-View Face Detection and Facial Feature Extraction using Color, Shape and Symmetry based Cost Functions. *Pattern Recognition Letters*, 19(8):669–680, 1998.

[101] R. Sanchez-Yanez, E. Kurmyshev, and F. Cuevas. A Framework for Texture Classification using the Coordinated Clusters Representation. *Pattern Recognition Letters*, 24(1–3):21–31, 2003.

[102] W. Schwartz and H. Pedrini. Textured Image Segmentation Based on Spatial Dependence Using a Markov Random Field Model. In *IEEE International Conference on Image Processing*, pages 2449–2452, Atlanta, GA, USA, Oct. 2006.

[103] W. R. Schwartz, F. R. de Siqueira, and H. Pedrini. Evaluation of Feature Descriptors for Texture Classification. *Journal of Electronic Imaging*, 21(2):023016–1–023016–17, Apr./June 2012.

[104] J. Serra. *Image Analysis and Mathematical Morphology*. Academic Press, New York, NY, USA, 1982.

[105] M. Shin, K. Chang, and L. V. Tsap. Does colorspace transformation make any difference on skin detection? In *Sixth IEEE Workshop on Applications of Computer Vision*, pages 275–279, 2002.

[106] L. Sigal, S. Sclaroff, and V. Athitsos. Estimation and Prediction of Evolving Color Distributions for Skin Segmentation Under Varying Illumination. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 582–587, 2000.

[107] M. Singh and S. Singh. Spatial Texture Analysis: A Comparative Study. In *International Conference on Pattern Recognition*, pages 676–679, 2002.

[108] F. Siqueira, W. Schwartz, and H. Pedrini. Multi-Scale Gray Level Co-occurrence Matrices for Texture Description. *Neurocomputing*, 120:336–345, Nov. 2013.

[109] F. R. Siqueira, W. R. Schwartz, and H. Pedrini. Adaptive Detection of Human Skin in Color Images. In *IX Workshop de Visão Computacional*, pages 1–6, Rio de Janeiro, Brasil, June 2013.

[110] M. J. Swain and D. H. Ballard. Color Indexing. *International Journal of Computer Vision*, 7:11–32, 1991.

[111] M. Tahir, A. Bouridane, and F. Kurugollu. An FPGA Based Coprocessor for GLCM and Haralick Texture Features and their Application in Prostate Cancer Classification. *Analog Integrated Circuits and Signal Processing*, 43(2):205–215, 2005.

[112] H. Tamura, M. S, and Y. Yamawaki. Textural Features Corresponding to Visual Precepetion. *IEEE Transactions on Systems, Man, and Cybernetics*, 8:237–247, 1978.

[113] Z. Tao, H. Wenxue, and W. Jinjia. A Novel Texture Analysis Method Based on Graph Spectral Theory. In *Fifth International Conference on Intelligent Information Hiding and Multimedia Signal Processing*, pages 467 –470, Sept. 2009.

[114] A. Torralba, K. Murphy, W. Freeman, and M. Rubin. Context-Based Vision System for Place and Object Recognition. In *Ninth IEEE International Conference on Computer Vision*, pages 273–280, Oct. 2003.

[115] J. Y. Tou, Y. H. Tay, and P. Y. Lau. A comparative study for texture classification techniques on wood species recognition problem. In *Fifth International Conference on Natural Computation*, volume 5, pages 8–12, 2009.

[116] M. Tuceryan and A. K. Jain. Texture Analysis. In C. H. Chen, L. F. Pau, and P. Wang, editors, *The Handbook of Pattern Recognition and Computer Vision*, pages 207–248. World Scientific Publishing Co., Aug. 1998.

[117] O. Tuzel, F. Porikli, and P. Meer. Region Covariance: A Fast Descriptor for Detection And Classification. In *9th European Conference on Computer Vision*, pages 589–600, 2006.

[118] UMD. Dataset, 2011. `http://www.cfar.umd.edu/~fer/High-resolution-data-base/hr_database.htm`.

[119] M. Unser. Texture Classification and Segmentation using Wavelet Frames. *IEEE Transactions on Image Processing*, 4(11):1549 – 1560, Nov. 1995.

[120] V. Vezhnevets, V. Sazonov, and A. Andreeva. A survey on pixel-based skin color detection techniques. In *GraphiCon*, pages 85–92, 2003.

[121] VisTex. Dataset, 2011. `http://vismod.media.mit.edu/vismod/imagery/VisionTexture/vistex.html`.

[122] R. Walker, P. Jackway, and D. Longstaff. Genetic Algorithm Optimization of Adaptive Multi-Scale GLCM Features. *International Journal of Pattern Recognition and Artificial Intelligence*, 17(1):17–39, 2003.

[123] H. Wang and S. F. Chang. A highly efficient system for automatic face detection in MPEG video. *IEEE Transactions on Circuits and Systems for Video Technology*, 7(4):615–628, Aug. 1997.

[124] J. S. Weszka, C. R. Dyer, and A. Rosenfeld. A comparative study of texture measures for terrain classification. *IEEE Transactions on Systems, Man and Cybernetics*, SMC-6(4):269–285, 1976.

[125] B. Wu and R. Nevatia. Detection of Multiple, Partially Occluded Humans in a Single Image by Bayesian Combination of Edgelet Part Detectors. In *IEEE International Conference on Computer Vision*, pages 90–97, 2005.

[126] G. Xu and T. Sugimoto. Rits Eye: A software-based system for realtime face detection and tracking using pan-tilt-zoom controllable camera. In *International Conference on Pattern Recognition*, pages 1194–1197, Brisbane, Australia, Aug. 1998.

[127] Y. Xu, S. B. Huang, and H. Ji. Integrating Local Feature and Global Statistics for Texture Analysis. In *16th IEEE International Conference on Image Processing*, pages 1377 –1380, Nov. 2009.

[128] M.-H. Yang and N. Ahuja. Gaussian mixture model for human skin color and its applications in image and video databases. In *SPIE: Storage and Retrieval for Image and Video Databases*, San Jose, CA, USA, Jan. 1999.

[129] Youtube. Statistics, 2013. `http://www.youtube.com/yt/press/statistics.html`.

[130] B. Zarit, B. Super, and F. Quek. Comparison of Five Color Models in Skin Pixel Classification. In *International Workshop on Recognition, Analysis and Tracking of Faces and Gestures in Real-Time Systems*, pages 58–63, 1999.

[131] J. Zhang. Brief Review of Invariant Texture Analysis Methods. *Pattern Recognition*, 35(3):735–747, 2002.

[132] J. Zhang, S. Lazebnik, and C. Schmid. Local Features and Kernels for Classification of Texture and Object Categories: A Comprehensive Study. *International Journal of Computer Vision*, 73(2):213–238, June 2007.

[133] W. Zhang, S. Shan, W. Gao, X. Chen, and H. Zhang. Local Gabor Binary Pattern Histogram Sequence (LGBPHS): A Novel Non-Statistical Model for Face Representation and Recognition. In *10th IEEE International Conference on Computer Vision*, volume 1, pages 786–791, Oct. 2005.