Democratic and Popular Republic of Algeria

Ministry of Higher Education and Scientific Research

University of Kasdi Merbah, Ouargla

Faculty of New Information and Communication Technologies

Department of Computer Science and Information Technology

De Nouvelles Caractéristiques pour la Recherche d'Images

Dissertation Submitted to the Department of Computer Science and Information Technology in Candidacy for the Degree of "Doctor" 3rd Cycle LMD in Computer Science.

Presented by:	*Supervised by:*
Mr. Belal Khaldi	Dr. Mohammed Lamine Kherfi

Jury Members

KHOLLADI Mohamed Khireddine	Prf. President	El-Oued University
KAZAR Okba	Prf. Examiner	Biskra University
KORICHI Ahmed	MCA Examiner	Ouargla University
LAALLAM Fatima Zohra	MCA Examiner	Ouargla University

Academic Year: 2016 – 2017

Dedication

I dedicate this work especially to the world dearest persons to me, who inspired me and gave me courage and hope, to my parents.

With great pleasure and joy:

To my dear brothers and sister

To all my friends

To all those who were giving me any kind of support, a sincere smile ☺.

Acknowledgments

First of all, I would like to acknowledge the assistance of my supervisor Dr. M. L. Kherfi for his patience, support and guidance throughout the course of this research study. I am deeply indebted to him also for his prompt and constructive feedback to accomplish this dissertation.

Additionally, I would like to express my deep gratitude for the generous support of the chief of our doctoral programme and our beloved professor, Dr. A. Korichi.

I would also like to extend my special thanks to the honourable members of the jury who accepted to examine and evaluate this study namely: Prf. KHOLLADI Mohamed Khireddine from the University of El-Oued , Dr. KORICHI Ahmed and Dr. LAALLAM Fatima Zohra from the University of Ouargla ; Prf. KAZAR Okba from the University of Biskra.

I am particularly thankful to the contribution of my colleague O. Aiadi who made this study feasible through his cooperation and helpful advices.

Last but not least, I would like to thank with gratitude Mr. and Mss. Chriet for their helpful lectures in image processing. Additionally, special thanks go to Mr. hadjem for his great support, notes and feedback.

Abstract

Gray-Level Co-occurrence Matrix (GLCM) is one of the most used methods for texture representation. It is generated by calculating the occurrence frequency of gray-level pairs within a texture image. Because it can be computed only from gray-level images, a significant amount of information that could be provided by color is totally ignored. Many literature works attempt to integrate information about color into the GLCM. However, most of these works suffer from numerous weakness such as miss-integration of color and hard mapping. In this research, we propose a generalization of GLCM from gray-level to HSV color space, which we refer to as Modified Integrative Color Intensity Co-occurrence Matrix (MICICM). To reach such a generalization, a soft mapping function, which determines for each pixel value the bin it falls into, is needed. In the previous studies, this function uses a hard mapping where all pixel values that fall in a bin are considered as the same, regardless their values. This presents a number of drawbacks. To fix them, we introduce a soft color and gray-level mapping scheme based on a set of weight assignment functions we propose. In our scheme, each pixel is mapped to more than one possible color (and gray level) bin, in order to avoid the drawbacks of hard mapping. Experimental results demonstrate the effectiveness of our MICICM compared to those of the state of the art. The findings of this investigation complement those of earlier studies and make several noteworthy contributions to image representation.

Keywords: Co-occurrence matrix, GLCM, MICICM, image features, fuzzy mapping, texture.

Résumé

Gray-Level Co-occurrence Matrix (GLCM) est l'une des méthodes les plus utilisées pour la représentation de texture. Elle est générée en calculant la fréquence d'occurrence des paires de niveaux de gris dans une image de texture. Parce qu'elle peut être calculée uniquement à partir des images de niveaux de gris, une quantité importante d'informations qui pourraient être fournies par la couleur est totalement ignorée. Beaucoup des travaux de la littérature tentent d'intégrer des informations sur la couleur dans le GLCM. Cependant, la plupart de ces travaux souffrent de nombreuses faiblesses telles que mal-intégration de la couleur et du mappage dur. Dans cette recherche, nous proposons une généralisation de GLCM du niveau de gris à l'espace de couleur HSV, que nous appelons Modified Integrative Color Intensity Co-occurrence Matrix (MICICM). Pour parvenir à une telle généralisation, une fonction de mappage douce, qui détermine pour chaque valeur de pixel le bin il tombe dans, est nécessaire. Dans les études précédentes, cette fonction utilise un mappage dur où toutes les valeurs de pixels qui tombent dans un bin sont considérés comme les mêmes, quelle que soit leur valeur. Ceci présente un certain nombre d'inconvénients. Pour les corriger, nous introduisons un schéma basé sur le mappage doux de couleur et de niveau de gris basé sur un ensemble de fonctions d'affectation de poids que nous proposons. Dans notre système, chaque pixel est mappé à plus d'un bin de couleur possible (et niveau de gris), afin d'éviter les inconvénients du mappage dur. Les résultats expérimentaux démontrent l'efficacité de notre MICICM par rapport à ceux de l'état de l'art. Les résultats de cette investigation complètent celles des études antérieures et font plusieurs contributions remarquables à la représentation de l'image.

Mots clés : matrice de Cooccurrence, GLCM, MICICM, caractéristiques de l'image, le mappage flou, texture.

ملخص

تعد مصفوفة ال GLCM واحدة من أكثر التقنيات المستعملة في تمثيل صور التراكيب، ويمكن استخراج هاته المصفوفة عن طريح حساب معدل ظهور مستويين رماديين بشكل متتالي في الصورة، ولأنه يتم حسابها فقط باستعمال المستويات الرمادية في الصورة فأن كمية كبية جدا من المعلومات التي تصف الألوان يكون مصيرها الضياع عن استعمال هذه التقنية. قام عدة باحثون بمحاولة تضمين معلومات الألوان في هاته التقنية غير أن العديد منه هاته المحاولات تعاني من نقائص مختلفة مثل سوء تضمين الألوان او استعمال الاسقاط القاسي للألوان. انطلاقا من المستوي اللوني HSV، نقوم في هذا البحث باقتراح تعميم لهاته التقنية لتشمل الألوان. حتى يتحقق مثل هذا التعميم، يجب إيجاد صيغة تحدد لكل لون المجال الذي ينتمي اليه. تتبنى معظم الاعمال السابقة اسقاط صارم للألوان اين يتم اعتبار جميع الالوان، التي تنتمي الى مجال واحد، متطابقة بغض النظر عن قيمها، وهذا ما يؤدي الى جملة من النقائص. للتخلص من هاته المشاكل، نقوم في بحثنا بتقديم مخطط بهدف الاسقاط اللطيف للألوان والذي يقوم أساسا على مجموعة من الدوال الرياضية التي تسند لكل لون اوزان مختلفة، كل وزن من هاته الاوزان يحدد نسبة انتماء اللون الى مجال معين. تؤكد التجارب التي اجريناها في هذا البحث تفوق صيغتنا عن باقي الصيغ المقترحة في الاعمال الأخرى. النتائج المتحصل عليها في هذا البحث تكمل باقي الاعمال المنجزة في هذا السياق كما تقدم مجموعة من المساهمات الملحوظة في فن تمثيل الصور.

كلمات مفتاحية: *مصفوفة الظهور المتتالي، خصائص الصور، الاسقاط اللطيف، صور التراكيب.*

Table of Contents

Chapter VI. Experimentation and Validation. 91

Key to Abbreviations and Acronyms

BRISK:	Binary Robust Invariant Scalable Keypoints.
CBIR:	Content-based image retrieval.
CCH:	Cell Color Histogram.
CCM:	Color Co-occurrence Matrix.
DCT:	Distance Cosine Transform.
DCWT:	dual-tree complex wavelet transform.
DWT:	Discrete Wavelet Transform.
FCCM:	Fuzzy Color Co-occurrence Matrix.
FT:	Fourier Transform.
GGLDM:	Generalized Gray-Level Difference Method.
GLCM:	Gray Level Co-occurrence Matrix.
HMMD:	Hue-Max-Min-Diff.
HSI:	Hue, Saturation and Intensity.
HSV:	Hue, Saturation, and Value.
ICIM:	Integrated Color Intensity Co-occurrence Matrix.
L*a*b:	Luminance, a-axes and b-axes.
LBP:	Local binary Pattern.
LCH:	Local Color Histogram.
MAPER:	Multi-Atlas Propagation with Enhanced Registration.
MARS:	Multimedia Analysis and Retrieval System.
MCCM:	Modified Color Co-occurrence Matrix.
MCM:	Multi-channel Co-occurrence Matrices.

MIA:	Medical image analysis.
MICICM:	Modified Integrative Color Intensity Co-occurrence Matrix.
MRI:	Magnetic resonance Imaging.
NTSC:	National Television System Committee.
PCA	Principal Component Analysis
QBIC:	Query by Image Content System.
RGB:	Red, Green and Blue.
SCM:	Single-channel Co-occurrence Matrices.
SIFT:	Scale-Invariant Feature Transform.
SURF:	Speeded Up Robust Features.
TBIR:	Text-Based Image Retrieval.
WLD:	Weber Local Descriptor.
YIQ	Luma, In-phase and Quadrature

List of Figures

List of Tables

Chapter I. GENERAL INTRODUCTION

A digital image is a two or three-dimensional numerical representation of a physical scene or interior structure of an object. This representation comes in form of a fixed-size array of real or complex numbers. Each array's element represents the values that describes the corresponding pixel (two-dimensional image) or voxel (three-dimensional image). Digital images hold meaningful information that can be extracted using the means of image processing techniques. These techniques such as edge detectors (Shrivakshan and Chandrasekar 2012) or neural networks (Rojas 2013) are inspired from human visual perception.

There are numerous of techniques that are used in automatic image analysing, each one could be useful to analyse the image from one or more aspect, e.g. object recognition (Grimson and Huttenlocher 1990) is used for localizing objects within an image, image segmentation (Morel and Solimini 2012) is for segmenting the image into a set of distinctive areas, motion detection (Borst and Egelhaaf 1989) is for detecting a moving object within an image, etc. However, there is no generic technique that cover all the aspects.

Digital image analysis is the arts of using these techniques to allow a machine (i.e., computer or some other electrical device) to, automatically, study an image to obtain the most meaningful information about its contents. These information could be, then, exploited to take a decision by a specialist or the machine itself.

I.1 Image Analysis Applications

It is known that computers are capable of analysing a large amounts of data in a short time. Moreover, computers can efficiently deals with tasks that require complex computation, or extract quantitative information. On the other hand, the human visual cortex can extract higher-level information, which makes it an excellent analysis tool. Therefore, Image analysis tries to invent perception models inspired by human vision trying to grasp image meanings as human does.

Because of the importance and the vital role that digital image analysis plays, its applications are continuously expending through almost all areas of science and industry, including:

I.1.1 Medicine

Medical image analysis (MIA) is the science that exploits different imaging modalities and digital image analysis techniques in order to solve/analyse medical problems. For example, Magnetic Resonance Imaging (MRI) is a safe medical imaging technique used to photograph the anatomy and physiological processes of the body in health and sickness. MIA uses machine vision and machine learning techniques to, automatically, identify and localize anomalies in human tissues using MRI images. Commonly, after imaging the brain, the image is then analysed by a specialist to localize damaged tissues. In MIA , however, segmentation techniques could be used to automate such a tedious process (Bandhyopadhyay and Paul 2012). Figure 1 shows the result of MRI brain image segmentation using Multi-Atlas Propagation with Enhanced Registration (MAPER) method as it is described in (Heckemann, Keihaninejad et al. 2010).

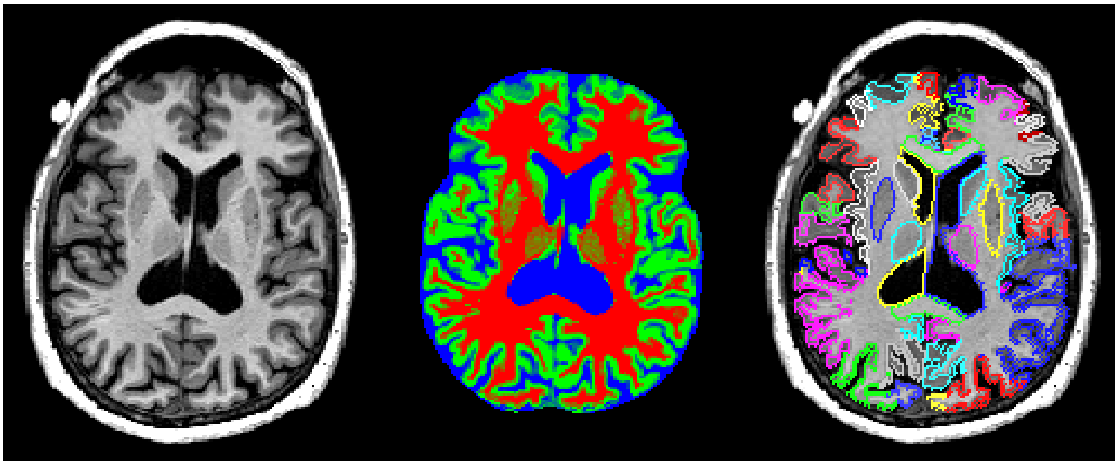


Figure 1. The result of MRI brain image segmentation, using MAPER method (Heckemann, Keihaninejad et al. 2010).

MIA can solve numerous problems such as, segmentation (i.e., delineating different organs), classification (e.g., determining types of leukocytes), registration (i.e., comparing different modalities/patients), reconstruction (e.g., making 3d-measurments), measuring flow (e.g., inside aorta), reconstructing flow fields (e.g., inside the heart), building shape priors efficiently, visualizing results, etc.

I.1.2 Security

In defense and security applications, Image processing approach aims to reduce the workload of human analysts in order to handle the ever-increasing volume of image data that is being collected. In addition, it aims to develop algorithms and approaches that will significantly assist to build fully autonomous systems capable of decisions and actions based only on sensor inputs. Because of this, advanced policies of security privacy such as fingerprint, eyeprint and facial detection are needed so that preclude mistaken identities and address frauds. An image of the finger, eye or face is taken then analysed to identify/authentify the users (Galbally, Marcel et al. 2014). Figure 2 shows an image of fingerprint after analysing and localizing all its significant components.

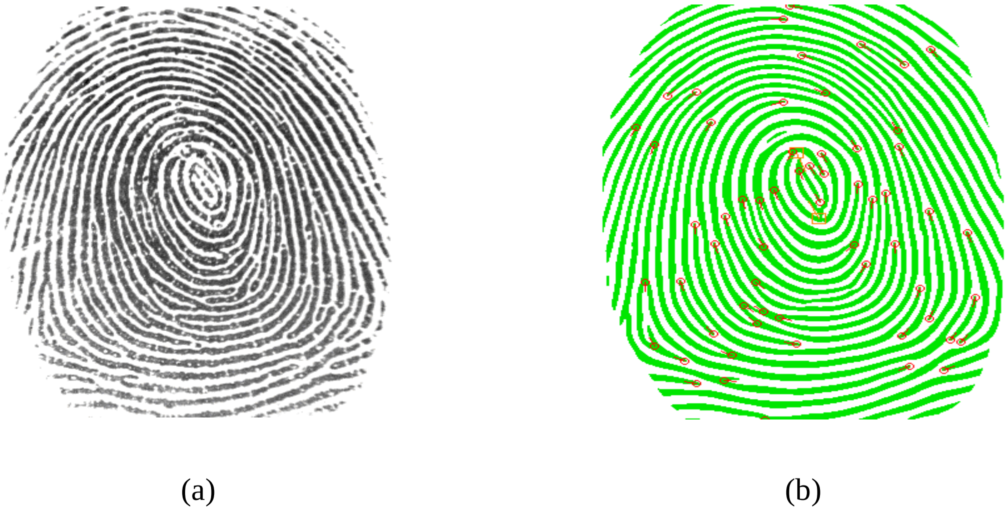


Figure 2. Fingerprint analysis. (a) Inputted fingerprint (b) outputted interest points that differs one print from others.

Image processing in defense and security can solve numerous problems such as, information assurance (e.g., watermarking and visual secret sharing protocols), steganography, target detection and tracking (e.g., abnormal event detection, human detection and tracking), and biometrics (e.g., iris recognition, face recognition and multicliet identification).

I.1.3 Microscopy

The main goal in imaging-based biological experiments is to extract structural, spatial and functional quantitative information about biological phenomenons accurately and automatically. For a given swab image, it is unlikely for human naked eye to count or define the properties of different germs. Microscopy image processing extracts precisely all useful information about the swab, such as number, size and types of different germs or truck their motions in order to define their behaviour pattern (Harder and Rohr 2013). Figure 3 shows an example of microscopic cell-split automatic tracking.

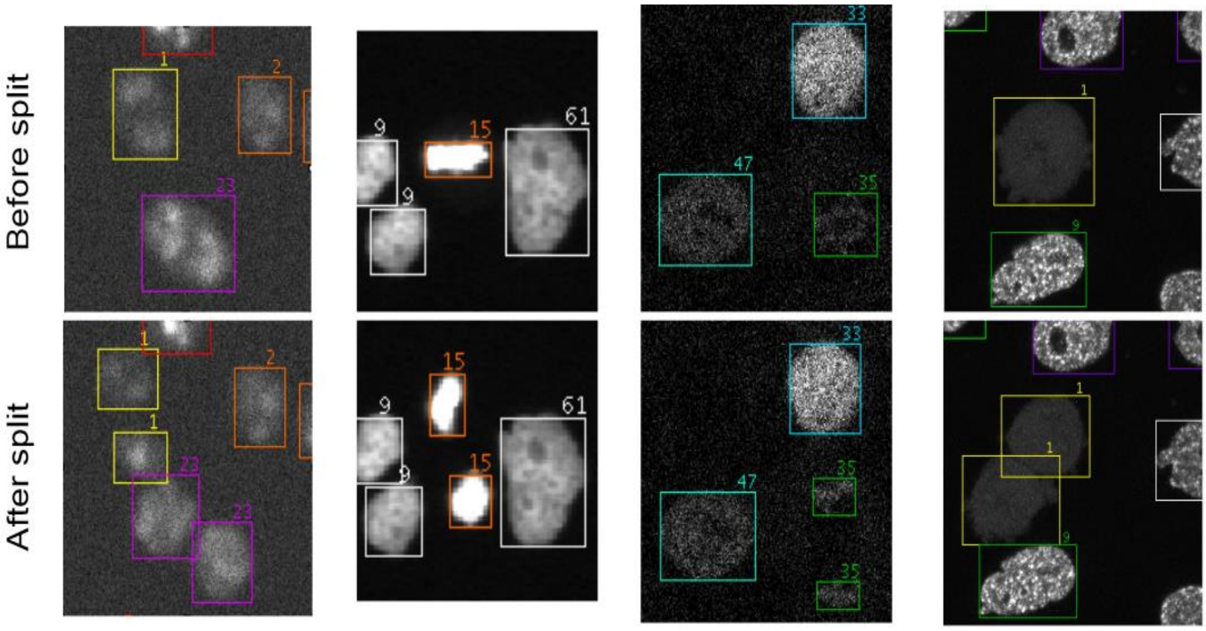
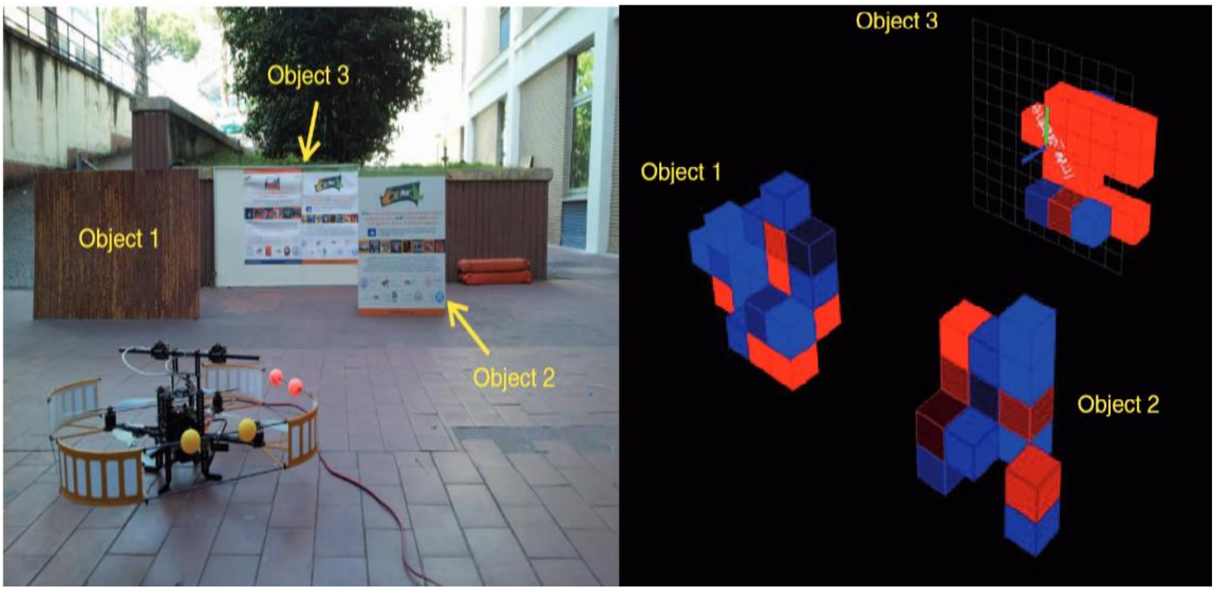


Figure 3. Examples of detected mitosis events in different types of image data(Harder and Rohr 2013).

Acoording to (Rittscher, Machiraju et al. 2008), microscopy image processing solves numerous problems such as, intelligent image acquisition, deconvolution, denoising and restoration; registration and mosaicing; segmentation, tracing, and tracking; classification and clustering; and modeling.

I.1.4 Robotics

Robots can be equipped with multiple vision sensors for better capability to compute the depth of a sense in the environment. By analysing the input image sequence, the robot can detect obstacle locations, types, and distances in order to construct an occupancy grid-map about the outdoor environment (Carloni, Lippiello et al. 2013). Figure 4 shows an example of an occupancy grid-map constructed after the robot has explored the environment.



(a) (b)

Figure 4. An example of an occupancy grid map of an outdoor environment that has been constructed by a robot (Carloni, Lippiello et al. 2013).

I.1.5 Retrieval

It is a tedious and very time-consuming task for a user that has an image or a sketch, to find all similar images from a big collection. By using image analysis techniques, this task become faster and easier. Content-based image retrieval (CBIR) (Veltkamp, Burkhardt et al. 2013) uses the content of the image rather than its metadata such as keywords or tags. Figure 5 depicts a general scheme of CBIR system where the input is an image example and the output is a set of images with similar content. Because this subject is particularly related to our research, we will discuss it, along with the techniques it uses, in details in Chapter IV.

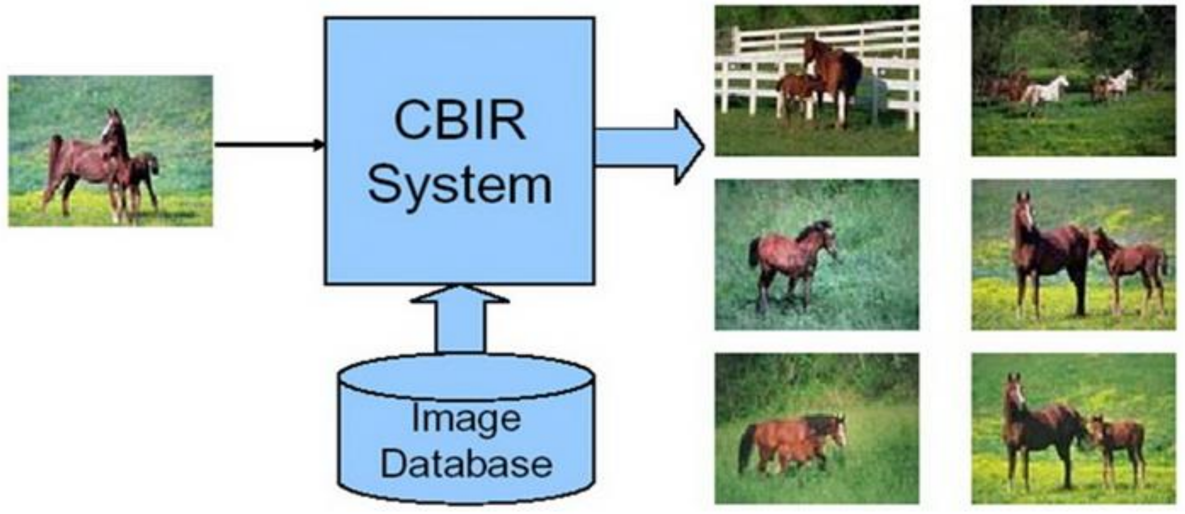


Figure 5. General Scheme of CBIR system.

These different uses of image analysis techniques have a main issue which is image representation. For example, CBIR system needs an image representation that works for a heterogeneous image dataset (e.g., landscapes, fruits, cities. etc.). A global information like quantities of different colors within the image could fully satisfy this task (Swain and Ballard 1991). However, fingerprint analysis needs a more sophisticated representation that localize the interest points within the image. Hence, we can see that there is no generic representation that works for all tasks,

I.2 Scope of the Research

In this research, we have focused on global (i.e., absolute) color-texture features, and the validation study was thus performed using these features in matching and texture retrieval tasks. Numerous color-texture features, that are robust to different image distortions, have been proposed in the literature. These features can be categorized into three main approaches namely statistical, structural, and frequency based. Features show best performance when they are compatible with the structures of texture presented in the image. Therefore, there is no one type of texture feature that is superior to the others in all cases. For the applications of matching and image retrieval, statistical features are generally used due to their simplicity and rapidity, and they will be the focus of this research. One of the oldest, most successful and used statistical feature in texture representation is Gray Level Co-occurrence Matrix(GLCM) (Haralick and Shanmugam 1973). Due to the numerous weaknesses that GLCM is suffers from, many literature works have suggested to improve it in terms of rapidity, robustness and generalization. Since that GLCM has been proposed to deal with gray-level textures, the scope of this research interests in generalizing it so it became able to deal also with color textures.

As we will see in Chapter III, many proposed GLCM generalizations in the literature are limited in terms of their robustness to different imaging conditions. For this reason, the developed Modified Integrative Color Intensity Co-occurrence Matrix (MICIM) in this research should comply with but not limited to the following criterions that should apply to all GLCM generalizations.

1. **Photometric Robustness:** The features should be robust to lighting direction, shadow and shading changes.
2. **Geometric Robustness:** The features should be invariance to viewpoint, zoom, and orientation variations.
3. **Generality:** should be robust to variations of color space in which the image is represented (i.e., color or gray level). Additionally, it must be applicable for different tasks such as matching, retrieval and classification.

The previous requirements improve the repeatability of texture features. It means that the same texture type can be detected repeatedly regardless the variation of imaging conditions. To validate a feature for those requirements, an appropriate dataset should be selected to perform a texture matching or texture recognition. To ensure greater credibility, the evaluation has been carried out using a well-known dataset that contains only texture images (see *VI.*1.1). Our main goal is to study other generalizations of the GLCM and compare them with our proposed generalization method under the same experimentation conditions set.

I.3 Contribution

Since there is no generic representation that works for all tasks, an appropriate image representation should be selected for each task. Recent works have tried to generalize some representation to cover more tasks. Generalizing a representation makes us avoid choosing the appropriate representation, increases the performance and reduces the calculation by considering only one representation instead of many.

Gray Level Co-occurrence Matrix (GLCM)(Haralick and Shanmugam 1973) is a widely used texture representation. However, it suffers from the fact that it can be used only with gray-level images. Using it along with color histogram(Swain and Ballard 1991) could partially generalize it and make it suitable for color images. Nevertheless, this solution lies the problem of selecting a weight for each feature, and increases the complexity. In this research, we propose a generalization of GLCM so it can be used with color images without resort to any other color representations. This representation which we refer to as Modified Integrative Color Intensity Co-occurrence Matrix (MICICM) (Khaldi and Kherfi 2016) holds information about both color and intensity of texture images. To reach such a generalization, a mapping function, which determine for each pixel-value the bin it falls into, is needed. In the previous studies, this function uses a hard mapping where all pixel values that fall in a bin are considered as the same, regardless of their values. This presents a number of drawbacks. To fix them, we introduce a new color and gray-level mapping scheme based on a set of weight assignment functions we propose. This scheme is based on a set of weight assignment functions that we define, where each pixel is mapped to more than one possible color (and gray-level) bin, in order to avoid the drawbacks of hard color mapping as we will explain hereafter. This is justified by the fact that most hue values in the Hue, Saturation, and Value (HSV) color space result from the mixing of more than one color. By using the MICICM, the image is compactly represented with few features that come in form of third order statistical moments, which make it suitable for real time systems. It can be used in different applications of image processing and pattern recognition.

In this research, we have developed a CBIR system based on numerous color/texture features including MICICM. In our System, Querying can be done simply by submitting an example image. The returned result is then appeared in form of sorted list from the most relevant image to the least one. We use this CBIR to evaluate and compare the results obtained by our MICICM against the other methods.

I.4 Thesis Outline

Chapter II: Image Features

In the second chapter, we introduce the different techniques of image representation. In addition, we review the most known features proposed in literature in recent years. For simplicity, these latters have been categorized into three main approaches namely, color, texture, shape and interest point features. Chapter II can be seen as a general literature review of related works.

Chapter III: Chapter IIIGLCM Generalization, related work

In this chapter, we present the literature works that are most relevant to ours. These works share a common aim that is attempting to generalize the GLCM from the gray-level space to color space. Because our research deals with two issues namely color sampling/mapping and color integration, the listed works has been categorized according to the issue they are trying to resolve. We explain in detail, discuss, and then criticize each work separately.

Chapter IV: Content-Based Image Retrieval

This research aims to propose a generalized representation of the texture based on the GLCM. This representation can be exploited in various image processing/analysis tasks such as content-based image retrieval systems. In this chapter, we introduce and define the CBIR and the benefits it offers. We firstly list the different types of query formulation in the CBIR, the features it uses, and the applications it might be used for. In addition, we list some CBIR prototypes that are operational and online for public.

Chapter V: Modified Integrative Color Intensity Co-occurrence Matrix

In Chapter V, we introduce our proposed generalization of GLCM untitled Modified integrative color intensity co-occurrence matrix. We start this chapter by analyzing the HSV color space that we use in our proposed representation. We explain the proposed color (resp., gray level) mapping formulas, then list the different steps that constitute our MICIM extracting scheme. Finally, we try to make this scheme clearer by applying it on an example.

Chapter VI: Experimentation and Validation

In this chapter, we evaluate, compare, and discuss our proposed method of texture representation. The comparison has been considered against other three works that aims to generalize the GLCM.

Chapter II. IMAGE FEATURES

In machine learning, pattern recognition(Aiadi, Khaldi et al. 2016, Aiadi and Kherfi 2016) and image processing, a feature could be defined as a vector of values that are derived from an initial big set of measured data (i.e., dimensionality reduction). Feature vectors are generally tends to be informative and non-redundant. A specific structure in the image can often be represented in different ways based on the selected feature. If we take *'Edge'* feature as an example then, each pixel within the image is described using a boolean that determines, for each pixel ,whether an edge is present or not (i.e., *'true'* for the presence of edge and *'false'* for the absence). Alternatively, we can use a richer representation that provides a certainty measures (e.g., orientation of the edge) instead of a simple boolean. Similarly, an image could be represented in terms of colors using values that describe the quantities of different colors within the image. In some cases, solving a complicated problem may require a more sophisticated feature with a higher level of details, however, such a feature causes more computation and time consuming.

Features can be used to facilitate the subsequent learning, generalization, matching, etc. Each of these features was intended to describe a specific aspect(Deselaers, Keysers et al. 2008). Color(Lu and Chang 2007), shape(Wu and Wu 2009) , texture(Jalaja, Bhagvati et al. 2005) and interest points such as Scale-Invariant Feature Transform (SIFT)(Lowe 1999), Binary Robust Invariant Scalable Keypoints (BRISK)(Leutenegger, Chli et al. 2011) and Speeded Up Robust Features (SURF)(Bay, Ess et al. 2008), are among the main aspects. Images features can be broadly categorized into four main categories namely, color, texture, interest point and shape features. However, a features vector suffers sometimes from the high dimensionality that prevent systems from executing real time tasks. Thus, a dimensionality reduction process that reduces the information quantity and conserves the representation quality might be needed to overcome this issue.

II.1 Color Features

Color is one of the most basic and used feature in image representation. It consists in extracting first order moments (e.g., appearance frequency of yellow pixels) from the image. There are three main approaches for color feature extraction presented in literature namely, global, fixed-size regions, and segmentation-based approach.

II.1.1 Global Approach

In this approach, images are represented using values that describe the global visual view. Since no pre-processing tasks are needed, features extraction in this approach is usually fast and sample. However, these features suffer from a common weakness which is their inability to describe the spatial distribution of colors within images. Most global-based color features encode the extracted information in form of a histogram. Examples of such features are global color histogram(Swain and Ballard 1991), cumulative color histogram(Stricker and Orengo 1995) and statistical color moments.

II.1.1.1 Global Color Histogram

Because of its stability, simplicity of calculation in addition to its invariance to rotation, translation and scale changes, global color histogram (Swain and Ballard 1991) is a widely used feature. It represent the quantity of colors in the image. In simpler word, each cell *H(i)* in a histogram represents the number of pixels with the value *i*. For a given Image *I*, the global color histogram *H* of this image can be calculated using Eq. (1).

$$GCH(k) = \sum_{i=1}^{N} \sum_{j=1}^{M} f(I[i,j]), \text{ where } f(x) = \begin{cases} 1 \; if \; I[i,j] = k \\ 0 \quad otherwise \end{cases} \quad (1)$$

II.1.1.2 Cumulative Color Histogram

Along the lines of color histogram, cumulative color histogram does not just count for each bin its occurrence frequency, but also sum it up with the occurrence frequency of all the smaller bins. It can be calculated using formula showed in Eq. (2).

$$CCH(k) = \sum_{i=1}^{N} \sum_{j=1}^{M} [\![f(I[i,j]) + \sum_{l=1}^{k-1} H(k))]\!] \text{ , where } f(x) = \begin{cases} 1 \; if \; I[i,j] = k \\ 0 \quad otherwise \end{cases} \quad (2)$$

II.1.1.3 Statistical Color Moments

Color moments(Flickner, Sawhney et al. 1995) are ones of the simple yet very effective features which assume that the content of the image can be interpreted as a probability distribution. Hence, this probability distribution could be described using a set of unique moments (e.g., mean, standard deviation, and skewness). Mean(μ), standard deviation(σ) and skewness(γ) can, respectively, be calculated using Eq. (3)-(5), where N is the width and M is the length of the corresponding image. $I[i,j]$ is a pixel from the image, which is located at the column i and the row j.

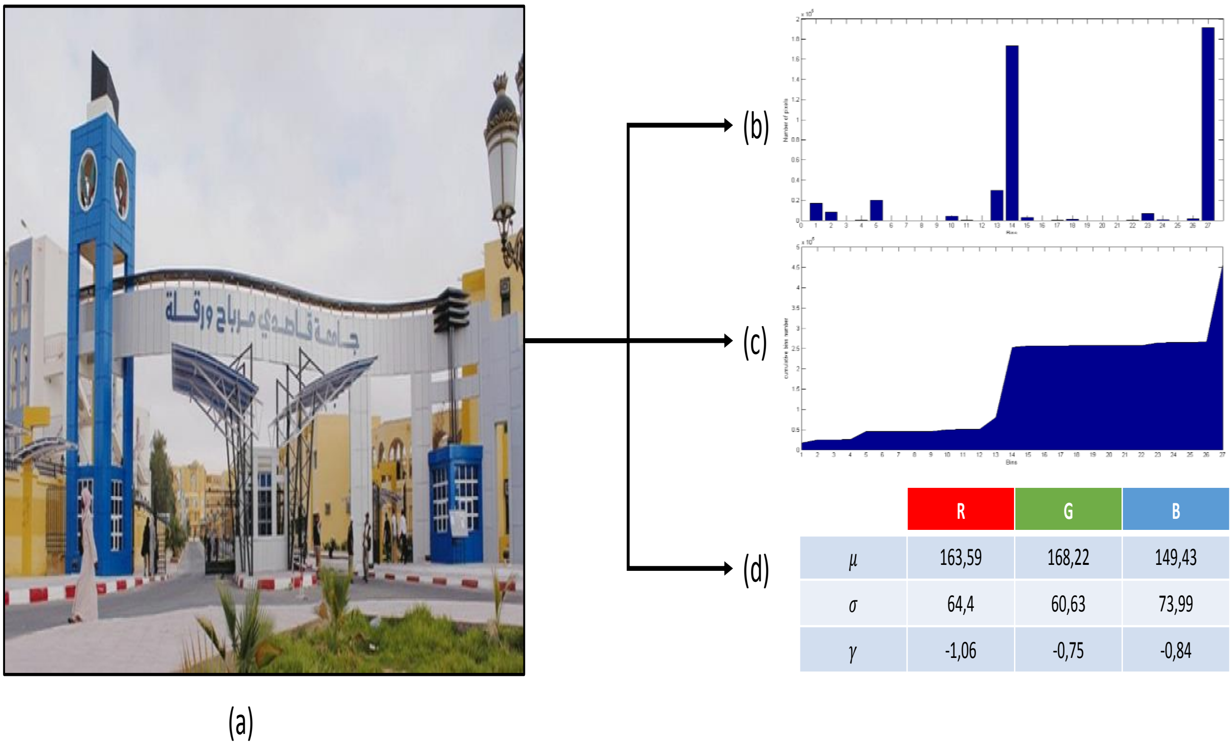
$$\mu = \frac{\sum_{i=1}^{N} \sum_{j=1}^{M} I[i,j]}{N \times M} \quad (3)$$

$$\sigma = \sqrt{\frac{\sum_{i=1}^{N} \sum_{j=1}^{M} (I[i,j] - \mu)^2}{N \times M}} \quad (4)$$

$$s = \sqrt{\frac{\sum_{i=1}^{N}\sum_{j=1}^{M}(I[i,j] - \mu)^3}{N \times M}} \qquad (5)$$

Many other color features have been introduced in literature, we mention: (Huang, Kumar et al. 1997, Manjunath, Ohm et al. 2001, Paschos, Radev et al. 2003, Chaira and Ray 2005, Nallaperumal, Banu et al. 2007, Williams and Yoon 2007).

Figure 6 shows an example of an image, and some of its corresponding color features.



	R	G	B
μ	163,59	168,22	149,43
σ	64,4	60,63	73,99
γ	-1,06	-0,75	-0,84

Figure 6. Extracting color features from an image where (a) is a given image, (b) is the extracted color histogram, (c) is the cumulative color histogram and (d) is a matrix of color moments.

II.1.2 Fixed-size Regions Approach

The fixed-size regions features divide, as a pre-processing step, the image into a set of cells that have a fixed size. Then, they extract color information from each cell separately. Finally, the extracted information are encoded in a single vector (generally, a large vector).

Examples of such features are Local Color Histogram(LCH)(Swain and Ballard 1991) and Cell Color Histogram (CCH)(Stehling, Nascimento et al. 2003).

II.1.2.1 Local Color Histogram

After dividing the image into a set of N equal blocks, LCH(Swain and Ballard 1991) extracts for each region its color histogram. Then, it combines these N histograms into one single histogram. LCH includes information about distribution of colors within different regions. The dissimilarity between two images is calculated by summing up the GCH distances between all possible pair of blocks (one of them in the first image and the other in the second). LCH costs usually in generating a very large vector.

II.1.2.2 Cell Color Histogram

In order to minimize the size of LCH, cell color histogram(Stehling, Nascimento et al. 2003) suggests to generate a histogram of regions instead of colors. It calculates for each color, the number of regions in which this color appears (the size of the vector is equal to the number of colors in the color space).

Many other features have been proposed in this context (Swain and Ballard 1991, Manjunath, Ohm et al. 2001, Li 2003, Sun, Zhang et al. 2006).

II.1.3 Segmentation-based Approach

Unlike Fixed-size regions, segmentation based approach divides the images into a set of regions that may vary from one image to another. This division is usually made as a pre-processing step using segmentation, pixel classification or clustering algorithms, which increases the complexity of this approach. Although they are more complex, features based on this approach offer better effectiveness. Many segmentation-based features have been introduced in literature (Pass, Zabih et al. 1997, Deng, Manjunath et al. 2001, Manjunath, Ohm et al. 2001, Stehling, Nascimento et al. 2001, Stehling, Nascimento et al. 2002, Lee, Lee et al. 2003, Palm 2004, Wong, Po et al. 2007, Yang, Chang et al. 2008, Chen, Jia et al. 2012).

It appears that color features are not enough to fully-represent the content of images. Therefore, color distribution could provide more meaningful information than color occurrence frequency. A sufficient representation of image content could be reached by combining color with texture features.

II.2 Shape Features

Humans consider Shape as an important cue for identifying and recognizing real-word object. Shape features look for an effective way to represent object based, either, on its exterior boundary information (i.e., contour based) or the interior region (i.e., region based) that the object occupies in the image(Zhang and Lu 2004). According to (Yang, Kpalma et al. 2008), an efficient shape feature must has some essential properties such as:

- **Identifiability:** perceptually similar objects must have the same shape features that differ from others.
- **Translation, rotation and scale invariance:** the location, the rotation or the scale changes of the object within the images must not affect shape features.
- **Affine invariance:** an affine transform can be constituted using sequences of translations, scales, flips, rotations and shears. Shape features must be as invariant as possible with affine transforms.
- **Noise resistance:** shape features must be highly robust against noise.
- **Occultation invariance:** shape feature of a part of an object must not change compared to the one that is extracted from full object.
- **Statistically independent:** two different shape features must be statistically independent (i.e., uncorrelated).
- **Reliability:** dealing with the same patterns means extracting the same shape features.

Many techniques have been proposed to represent shape features, Yang et al.(Yang, Kpalma et al. 2008) have organized the existing techniques of shape-based features extraction as it is illustrated in Figure 7.

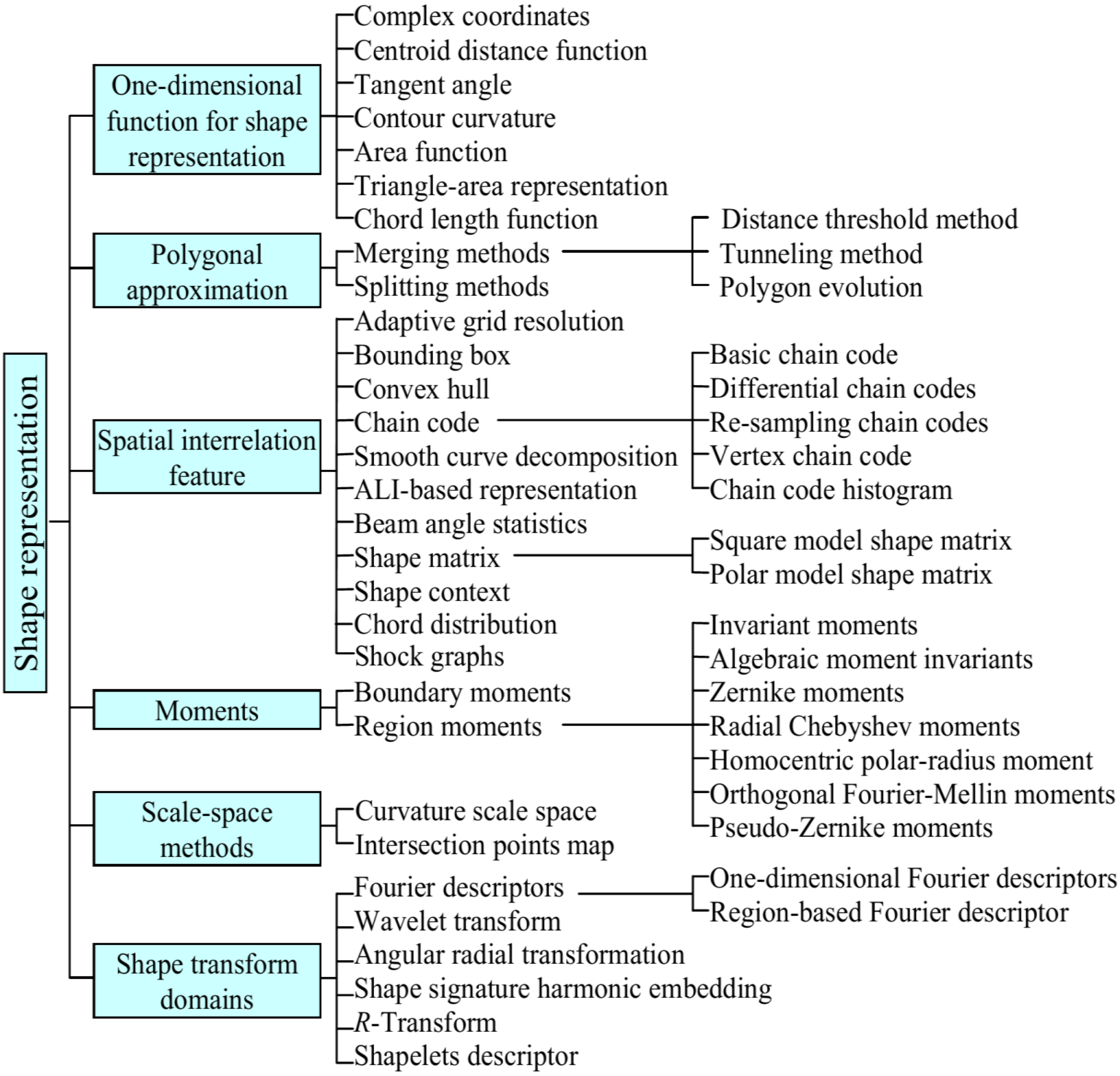


Figure 7. An overview of shape description techniques(Yang, Kpalma et al. 2008).

In contrast to the significant progress made in colour and texture representation, extracting shape features still an open issue due to the inherent complexity. In particular, it must identify regions occupied by an object in order to describe its shape. A number of known segmentation techniques combine low-level features with region-growing or split-and-merge processes to fulfil such a task. Generally, it is extremely hard to precisely segment an image into meaningful regions using low-level features due to: shadows, occlusions, non-uniform illumination, reflectivity, etc.

After segmenting different objects within a surface, their shapes (resp., forms) have to be described, indexed, and then compared. However, capturing all the aspects of visually perceived shapes as well as their comparison is a hardly possible process, and there is therefore no mathematical description able to accomplish this. This is because the elusive nature of shapes that hinder any formal analysis trading-off between the complexity of characterising shapes and the capability to describe and compare shapes of interest. Indeed, objects within an image can be characterised using either contour-based or region-based techniques or can also be a combination of the two.

II.2.1 Boundary-Based Representation of Shape

In boundary-based representation approach, shapes are described by closed curves surrounding them. These curves can be specified in numerous ways such as, chain codes, polygons, circular arcs, splines, or boundary Fourier descriptors. Figure 8 shows some examples of boundary-based representation techniques.

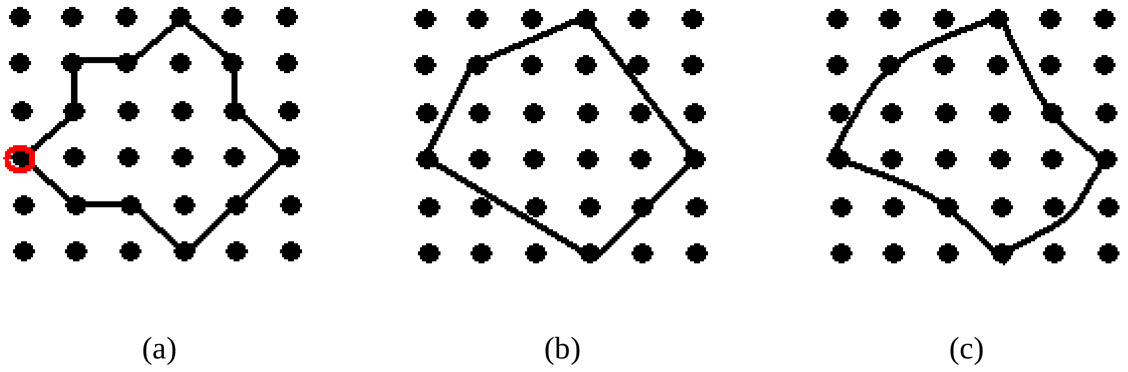


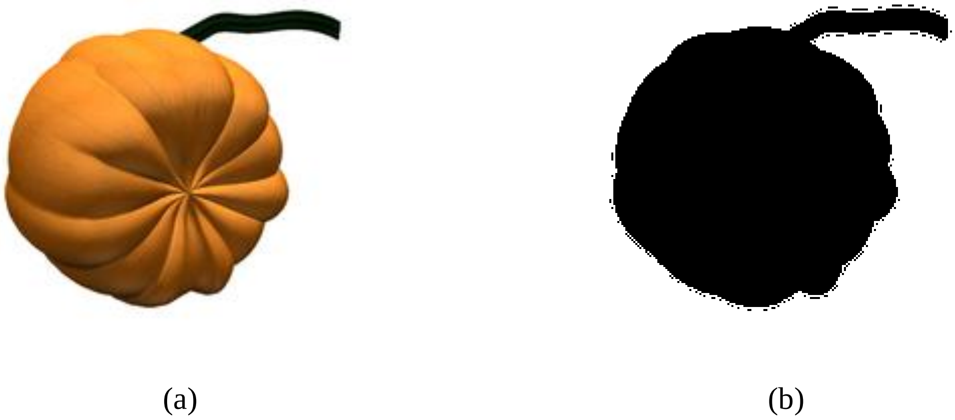
Figure 8. Some techniques of boundary-based representation of shape. (a) Chain code 707113235465, (b) polygonal approximation, and (c) arc-based approximation.

Many authors used chain code descriptor for various applications. It represents the object boundary by a sequence of integer, where each integer describes a one line from the boundary, which has a specified length and direction. Chain code provides a storage efficient representation of the object's boundary(Yang, Kpalma et al. 2008).

Alternatively, the boundary features could be represented as an ordered polygonal approximation. This technique tries to deform a pre-defined sketch set to adjust it to the shape of the target models.

II.2.2 Region-Based Representation of Shape

Region-based representation consists in specifying the body of an object within the closed boundary (interior description) as it is illustrated in Figure 9. The specified body is then represented using moment invariants, primitives collection (e.g., rectangles, disks, quadrics, etc.), deformable templates, skeletons, or simply a set of points.



(a) (b)

Figure 9. Example of region based segmentation where (a) is the original object, and (b) is the interior area of the object

By using *skeletons,* each shape could be represented with an axis of symmetry between a chosen pair of boundaries. One of the simplest skeletons is given by the *medial axis* that is defined as the trace of a locus of inscribed maximum-size circles. Usually, a skeleton is represented by a graph.

Another variant of the medial axis is the propagation from the boundaries, which creates a *shock set* (similar to a "grassfire" initiated from the boundaries). Shocks refers to singularities that are created from fronts propagating collisions. Thus, a shock graph is formed by adding shock dynamics to each point and grouping the monotonically flowing shocks into branches. Compared to the skeletal one, the shock graph produces a finer partition of the medial axis.

As shape is also defined in terms of presence and distribution of oriented parts, the quantitative characteristics of objects within an image (e.g., angular spectra of image components or edge directionality) may be used as global shape descriptors. For example, the blobworld model replaces each object with a blob, or an ellipse (identified by its centroid and scatter matrix) in order to efficiently describe objects separated from the background. Each blob is also characterised with two additional measures which are the mean texture and the two dominant colours. Expectation - Maximisation (EM) algorithm is used to segment the initial images based on colour and texture features.

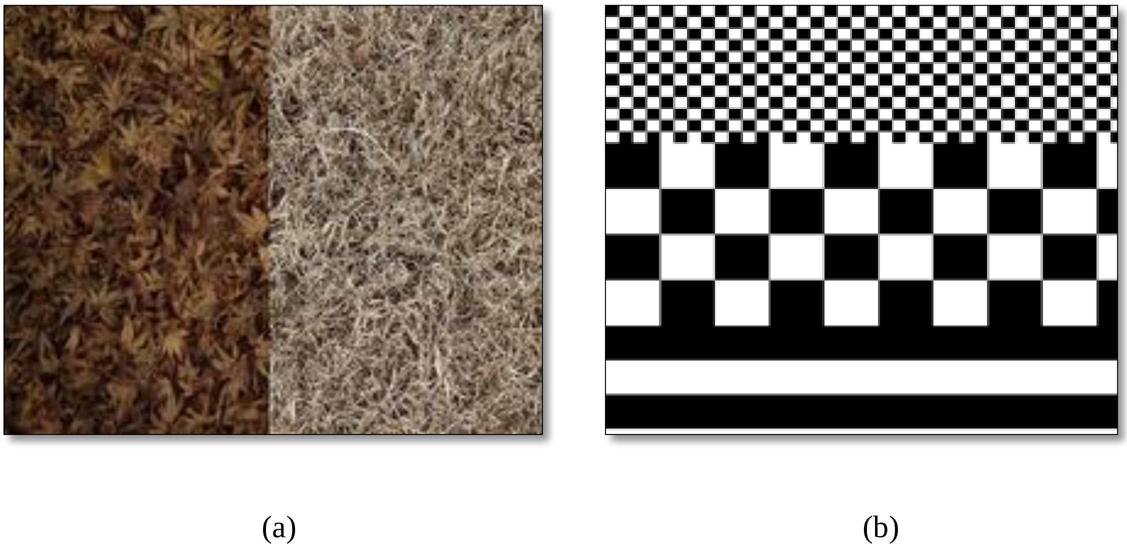
II.3 Interest Point Features

Interest point is a recent terminology in computer vision, which concerns in localizing some points with some specific properties in the image (Lowe 1999, Dalal and Triggs 2005, Calonder, Lepetit et al. 2008, Calonder, Lepetit et al. 2010, Rublee, Rabaud et al. 2011, Alahi, Ortiz et al. 2012). According to (Lindeberg 2013, Lindeberg 2013),an interest point could be characterized as follows:

- It has a clear, preferably mathematically well-founded, definition.
- It has a well-defined position in the image.
- Its local pixel-neighbourhood is rich of information contents.
- It is robust to local and global perturbations in the image (e.g., brightness/ illumination changing).

II.4 Texture Features

Texture refers to visual patterns that have properties of homogeneity and does not result from the presence of one single color or intensity(Smith and Chang 1996). Typical examples of natural texture could be found in clouds, trees, bricks, grass, flowers, etc. Texture remains a poorly defined concept (Tuceryan and Jain 1993). Figure 10 shows an example of two types of texture.



(a) (b)

Figure 10. Example of two types of texture, (a) natural texture and (b) artificial texture.

Texture can be easily recognized when it is seen, however, it is very difficult to define. This difficulty can be attributed to the various texture definitions given by vision researchers. We list some of these definitions as follows.

a. "We may regard texture as what constitutes a macroscopic region. Its structure is simply attributed to the repetitive patterns in which elements or primitives are arranged according to a placement rule."(Tamura, Mori et al. 1978).
b. "A region in an image has a constant texture if a set of local statistics or other local properties of the picture function are constant, slowly varying, or approximately periodic."(Sklansky 1978).
c. "The image texture we consider is nonfigurative and cellular... An image texture is described by the number and types of its (tonal) primitives and the spatial organization or layout of its (tonal) primitives... A fundamental characteristic of texture: it cannot be analyzed without a frame of reference of tonal primitive being stated or implied. For any smooth gray-tone surface, there exists a scale such that when the surface is examined, it has no texture. Then as resolution increases, it takes on a fine texture and then a coarse texture."(Haralick 1979).
d. "Texture is defined for our purposes as an attribute of a field having no components that appear enumerable. The phase relations between the components are thus not apparent.

Nor should the field contain an obvious gradient. The intent of this definition is to direct attention of the observer to the global properties of the display — i.e., its overall “coarseness,” “bumpiness,” or “fineness.” Physically, nonenumerable (aperiodic) patterns are generated by stochastic as opposed to deterministic processes. Perceptually, however, the set of all patterns without obvious enumerable components will include many deterministic (and even periodic) textures.”(Richards and Polit 1974).

e. “Texture is an apparently paradoxical notion. On the one hand, it is commonly used in the early processing of visual information, especially for practical classification purposes. On the other hand, no one has succeeded in producing a commonly accepted definition of texture. The resolution of this paradox, we feel, will depend on a richer, more developed model for early visual information processing, a central aspect of which will be representational systems at many different levels of abstraction. These levels will most probably include actual intensities at the bottom and will progress through edge and orientation descriptors to surface, and perhaps volumetric descriptors. Given these multi-level structures, it seems clear that they should be included in the definition of, and in the computation of, texture descriptors.”(Zucker and Kant 1981).

f. “The notion of texture appears to depend upon three ingredients: (i) some local‘order’ is repeated over a region which is large in comparison to the order’s size,(ii) the order consists in the nonrandom arrangement of elementary parts, and (iii) the parts are roughly uniform entities having approximately the same dimensions everywhere within the textured region.”(Hawkins 1970)

Since there is no agreement in defining the texture as a concept, we rather define it as the benefits it ensures. “Texture features provide important information about patterns and the relationship between them within a surface”. For this reason, texture is a very important and useful feature in image retrieval, recognition and classification. Several approaches have been proposed to represent texture. We mention the three most known ones which are statistical, frequency and geometrical.

II.4.1 Statistical Methods

Statistical methods are based on counting the occurrence frequency of some specific patterns within a texture image. For example, counting the occurrence frequency of color pairs such as red-blue, white-black, etc. A large number of statistical features have been proposed in literature starting from first order statistics to higher order ones (Haralick and Shanmugam 1973, Unser 1986, Kovalev and Volmer 1998, Tao and Dickinson 2000, Carkacioglu and Yarman-Yural 2001, Manjunath, Ohm et al. 2001, Çarkacıoğlu and Yarman-Vural 2003, Zhou, Xin et al. 2003, Hadjidemetriou, Grossberg et al. 2004, Huang and Liu 2007, Kiranyaz, Ferreira et al. 2008).

II.4.1.1 Tamura Features

After he proposed a set of specifications for texture, Tamura(Tamura, Mori et al. 1978) has showed them to pattern recognition invistigators in order to determine which ones among them are the common to all texture. Six of them fullfil this comnuness condition, which are:

1. *Coarseness (Coarse versus Fine):* coarseness is one of the fundamental features of texture. It refers to the scale changes between patterns. When two patterns differ only in scale, the magnified one is coarser.
2. *Contrast (High versus Low contrast):* The contrast of a given image can be measured by converting it to gray level, such that each value represents the contrast of its corresponding pixel. For given two images, the image with sharp edges (i.e., contrast rapidly changes) has higher contrast.
3. *Directionality (Directional versus Nondirectional):* Directionality describes the global property of a texture region. It stands for measuring the total degree of directionality of patterns within a given area. Patterns that belong to the same area will have the same directionality even if they differs in orientation.
4. *Line-likeness (Line versus Blob-like):* this feature is concerned with the shape of elements that constitute a texture. In contrast to Blob-like, Line-like feature describes the sharpness of the edges in texture images.
5. *Regularity (Regular versus Irregular):* This feature describes the variation rule that has been used for patterns placement. Since they have no regular placement rule, natural textures tend to be Irregular. Conversely, fine texture seems to be perceived as regular.
6. *Roughness (Rough versus Smooth)*: This feature is meant to be used with tactile textures, and not for visual ones. However, human perception is usually able to visually distinguish rough textures from smooth ones.

Tamura has approximately formulated those features mathematically based on the judgement of human subjects.

II.4.1.2 Laws Texture Energy Measures

An alternative approach for measuring texture energy is by using local masks to detect various types of patterns that constitute a texture. Laws(Laws 1980) has developed method that measures the energy of texture. He define different types of masks for the image to be convoluted with. These masks are a result of multiplying one vector with another from the following ones.

$L5\ (Level) = [\ 1 \quad 4 \quad 6 \quad 4 \quad 1\ ]$: Gives the centre-weighed local average.

$E5\ (Edge) = [\ -1 - 2 \quad 6 \quad 1 \quad 2\ ]$: For detecting region edges.

$S5\ (Spot) = [\ -1 \quad 0 \quad 2 \quad 0 \quad -1\ ]$: For detecting spots or isolated points.

$R5\ (Ripple) = [\ -1 - 2 \quad 6 \quad 1 \quad 2\ ]$: Detects ripples and waves.

The outer product of all possible two vectors (24 = 16) gives sixteen masks. For example, the mask L5E5 is calculated by multiplying L5 and E5 as follows:

$$\begin{bmatrix} 1 \\ 4 \\ 6 \\ 4 \\ 1 \end{bmatrix} \times [-1 - 2 \ \ 6 \ \ 1 \ \ 2\ ] = \begin{bmatrix} -1 & -2 & -6 & 1 & 2 \\ -4 & -8 & -24 & 4 & 8 \\ -6 & -12 & -36 & 6 & 12 \\ -4 & -8 & -24 & 4 & 8 \\ -1 & -2 & -6 & 1 & 2 \end{bmatrix}$$

Firstly, Laws convolve the image with a *Mean* mask in order to remove the effects of illumination. Then, he applied each of the masks on this pre-processed image, producing sixteen filtered images. Finally, he extracts the energy map for each of the filtered images according to Eq. (6). Where, F_k is the result of filtering a given image with the k^{th} mask. $F_k[i, j]$ Is the value of the pixel with the index *(i, j)* from this filtered image.

$$E_k[r,c] = \sum_{j=c-7}^{c} \sum_{i=r-7}^{r} |F_k[i,j]| \qquad (6)$$

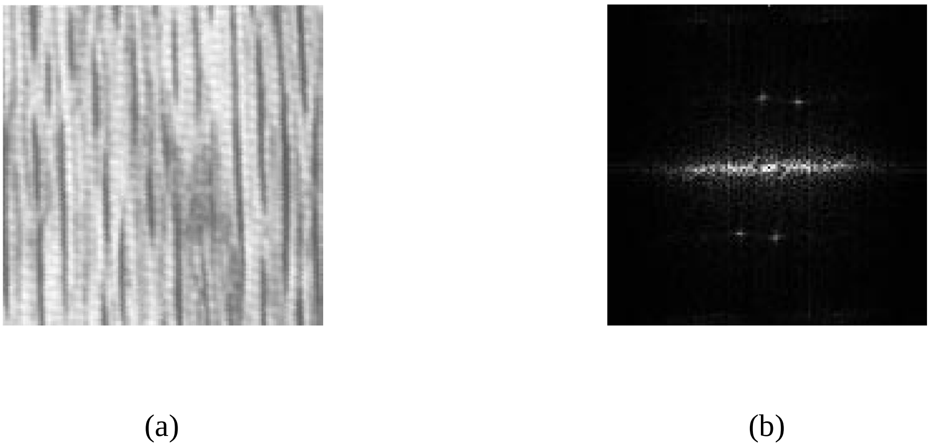
Thus, the image shall be represented with nine final maps. We said nine instead of sixteen because some energy maps are combined due to their symmetry. For example, L5E5 and E5L5 which respectively describes vertical and horizontal edge. The final energy maps can be used later on to cluster the image into regions of a uniform texture.

II.4.1.3 Autocorrelation Features

The repetitive nature of placement of texture elements in the image is a very important characteristic of many textures. The amounts of texture regularity, fineness and coarseness present in the image can be assessed using autocorrelation function. The autocorrelation function $\rho(x,y)$ of each transmission (x,y) of a given image I is formally defined as follows:

$$\rho(x,y) = \frac{\sum_{u=0}^{N} \sum_{v=0}^{N} I(u,v) I(u+x, v+y)}{\sum_{u=0}^{N} \sum_{v=0}^{N} I(u,v)^2} \qquad (7)$$

Autocorrelation function is highly related to the size of the texture primitive (i.e., textons). For a coarse texture (large size primitives), then the autocorrelation function will drop off slowly; otherwise (small size primitives), it will drop off very rapidly. This function will exhibit peaks and valleys for regular textures. It drops off and rises again in periodic manner as the primitives are spatially periodic. The relationship between the autocorrelation and the power spectrum function is therefore that they are Fourier transforms of one another(Yaglom 2004). Consider an image I in the spatial domain and its Fourier transform $F(u,v)$. Then, the quantity $|F(u,v)|^2$, where $|x|$ refers to the modulus of a complex number x , is defined as I's power spectrum. Figure 11 illustrates the power spectrum extracted from a texture image, it also shows how the directionality of a texture effects the distribution of energy.



(a) (b)

Figure 11. Extracting the power spectrum from a given image. (a) A direction-full texture image, and (b) its power spectrum that reflects in the directional distribution of energy.

II.4.1.4 Co-occurrence Matrices

Gray-Level Co-occurrence Matrix (GLCM)(Haralick and Shanmugam 1973) is amongst the well-known and widely used techniques for texture representation. This 2D matrix accumulates a set of second order statistics, which measures the spatial dependency of two gray levels given a displacement vector$(\Delta x, \Delta y)$. A set of texture features such as energy, contrast, entropy, homogeneity and correlation are then derived from the co-occurrence matrix. they holds statistics about the co-existence of some values at a given offset. However, the GLCM suffers from a number of shortcomings. For example, it appears that there is no general solution for optimizing the displacement vector(Chen, Pau et al. 1993). Usually, the gray-level space is quantized in order to reduce the size of the matrix and keep it manageable. More details will be given in Chapter III about this approach of representation.

II.4.2 Frequency Methods

Unlike spatial domain, frequency domain defines the texture as a set of signals with a given frequency, amplitude and direction. Frequency-based methods, firstly, transform the image into a set of frequency bands (e.g., by using Fourier Transform (FT)), then, they extract the energy feature carried by each of those bands resulting in feature vector. This vector could be used, later on, to identify images. Many frequency-based methods have been proposed in literature (Manjunath and Ma 1996, Rubner and Tomasi 1999, Ro and Kang 2000, Wu, Manjunath et al. 2000, Sim, Kim et al. 2001, Yang and Liu 2002, Wang, Feng et al. 2004, Lee and Chen 2005, Han and Ma 2007, Janney and Yu 2007, Montoya-Zegarra, Beeck et al. 2008).

II.4.2.1 Gabor Filters-based Features

Gabor filters-based features extraction(Daugman 1980, Daugman 1985) is a popular frequency-based method. It uses a predefined filter set (i.e., filter bank) that varies in scale and orientation in order to define the different types of textures. In the spatial domain, a 2D Gabor filter is defined as a Gaussian kernel modulated by a complex sinusoidal plan as follows:

$$G(x,y) = \frac{f^2}{\pi\gamma\eta}\exp\left(-\frac{x'^2+\gamma^2y'^2}{2\sigma^2}\right)\exp(j2\pi fx'+\phi) \tag{8}$$

$$x' = xcos(\theta) + ysin(\theta) \tag{9}$$

$$y' = -xsin(\theta) + ycos(\theta) \tag{10}$$

where $f$and θ are dedicated, respectively, to specify the desired frequency and orientation of the corresponding filter.

By varying the values of the different parameters, a number of filters could be generated. Figure 12 shows an example of a filter-bank (40 filters) generated using different orientations and frequencies (8 orientations and 5 frequencies).

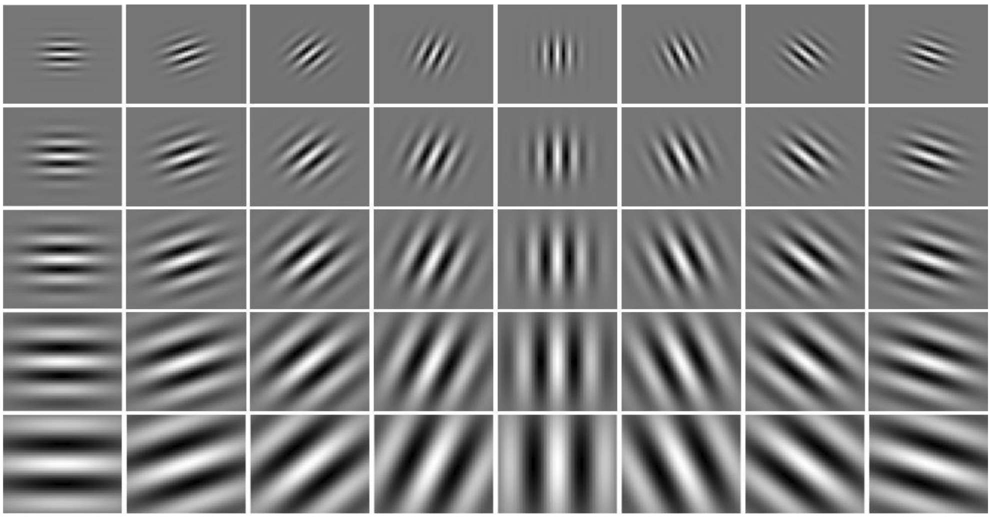


Figure 12. A set of 40 Gabor filters extracted using different frequencies and orientations.

After convoluting the image with a selected set of filters, various statistical measures could be extracted from the resulted images. For example, the mean and the variance are two features amongst many others.

Gist is a specific type of Gabor filter-based features. It convolves the image with 32 predefined filters that vary for 4 scales and 8 orientations. This process produces 32 feature maps from the original image. Then, each feature map is divided into 16 equal regions (4x4 grid). After concatenating the values' averages of these regions for all maps, we get a 512 (32x16) length Gist descriptor.

Robustness to scale, orientation, translation and image noise make Gabor filter-based features very powerful. However, they suffer from the high computation. This last drawback is resulted from the big size of feature vector that is extracted from each image. For example, if we have an image I with dimensions of 400×400, and we want to extract features based on filter bank that contains 40 filters, then its vector size will be $400 \times 400 \times 40 = 6400000$ elements.

II.4.2.2 Discrete Wavelet Transform Features

For a given data vector with a length equal to n^2 (such that $n \in N)$, the discrete wavelet transform (DWT) is a linear operator that transforms this vector into a numerically different vectors with the same length. In other words, DWT is a tool that separates the data into different frequency components, and then studies separately each component based on the resolution matched to its scale.

As it is illustrated in Figure 13, DWT decompose the image, using a cascade of filtering followed by a factor 2 subsampling, into:

1. Coarse approximation by convolving it with low-pass filter.
2. Detail information by convolving it with high-pass filter.

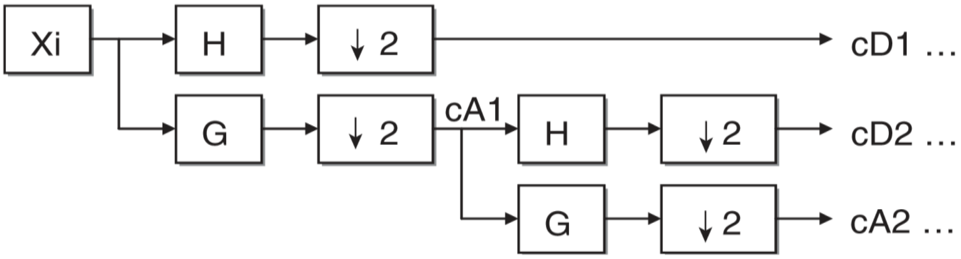


Figure 13. DWT fast algorithm schematic diagram, where H and G respectively are high-pass and low-pass filters, and ↓2 denotes down-sampling.

the outputs of these filters are given by the equations Eq. ((11)) and (12)), where *h(n)* and *l(n)* are two coefficients of low and high-pas filters respectively.

$$cD_{j+1}(p) = \sum_{n=-\infty}^{+\infty} h(n-2p)\, cD_j(n) \tag{11}$$

$$cA_{j+1}(p) = \sum_{n=-\infty}^{+\infty} l(n-2p)\, cD_j(n) \tag{12}$$

This decomposition is performed recursively on low-pass approximation coefficients obtained at each level, until reaching the necessary iterations. Different direction could be considered with each decomposition , for example: 0^o (horizontal), 45^o (diagonal) and 90^o (vertical). Those coefficients are considered as features for the image from which they were extracted. Typical examples of such methods are Haar (Talukder and Harada 2010), Daubechies (Wang, Wiederhold et al. 1998) and the dual-tree complex (DℂWT)(Selesnick, Baraniuk et al. 2005) wavelet transforms.

II.4.3 Geometrical Methods

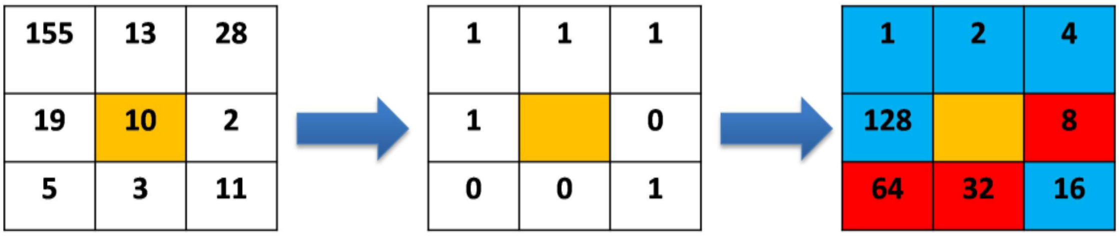
Geometrical methods aim to identify texture elements (i.e., primitives or textons or textels) that constitute a texture (Julesz 1981, Zhu, Guo et al. 2005, Fan, Li et al. 2008). These elements are arranged according to an arrangement rule. Thus, a texture could geometrically be defined as Eq. (13) shows.

$$f = R(e) \tag{13}$$

where f is the texture image, R is an arrangement rule and e is a texture element.

II.4.3.1 Local Binary Pattern

Local binary Pattern (LBP)(Ojala, Pietikäinen et al. 1996) has proven its ability as a powerful feature in many applications including texture classification and segmentation. The basic LBP operator is computed by assigning each pixel within the image to a binary code of eight bits. This code is computed by considering the center pixel of a neighborhood, with a given radius R, as a threshold value. The decimal value that corresponds to the binary one is then used to replace the original value of the center pixel. P is the set of neighbor pixels except the central one. For example, for a given R equals three, the appearance frequency of each code (i.e., pattern) in the image is then computed to constitute a histogram of 256 (2^8) dimensions. Figure 14 shows an example of the LBP calculation.



Sub-image

Pattern = 11101001

Replace the center pixel by
1 + 2 +4 +16 + 128 = 151

Figure 14. Example of extracting LBP value from a pixels neighborhood.

As a formula, the LBP code of a given pixel that is located at the point (x_c, y_c) is given by Eq. (14).

$$LBP_{P,R} = \sum_{p=0}^{P-1} s(g_p - g_c)2^p \ \ such\ that\ s(x) = \begin{cases} 1 & if\ x \geq 0 \\ 0 & otherwise \end{cases} \qquad (14)$$

By assuming that R equals three, there exist 58 amongst the 256 patterns provide more information than others do. This makes possible to use a smaller subset of patterns to describe image texture. These patterns are called uniform and contain at most two contiguous bit suits. Later on, LBP was extended to cover neighborhoods with different sizes(Ojala, Pietikainen et al. 2002).

II.4.3.2 Weber Local Descriptor

Weber Local descriptor (WLD) (Chen, Shan et al. 2008, Chen, Shan et al. 2010) consists in extracting, from each pixel, two components namely, differential excitation and orientation. Such that, the differential excitation component is a function that represents the ratio between the intensity of the current pixel and the relative intensity differences of this pixel against its neighbors. The orientation component is the gradient orientation of the corresponding pixel. Thus, WLD uses these two components to constitute a feature vector (i.e., histogram). For a given image *I*, Figure 15 depicts a WLD feature extraction scheme.

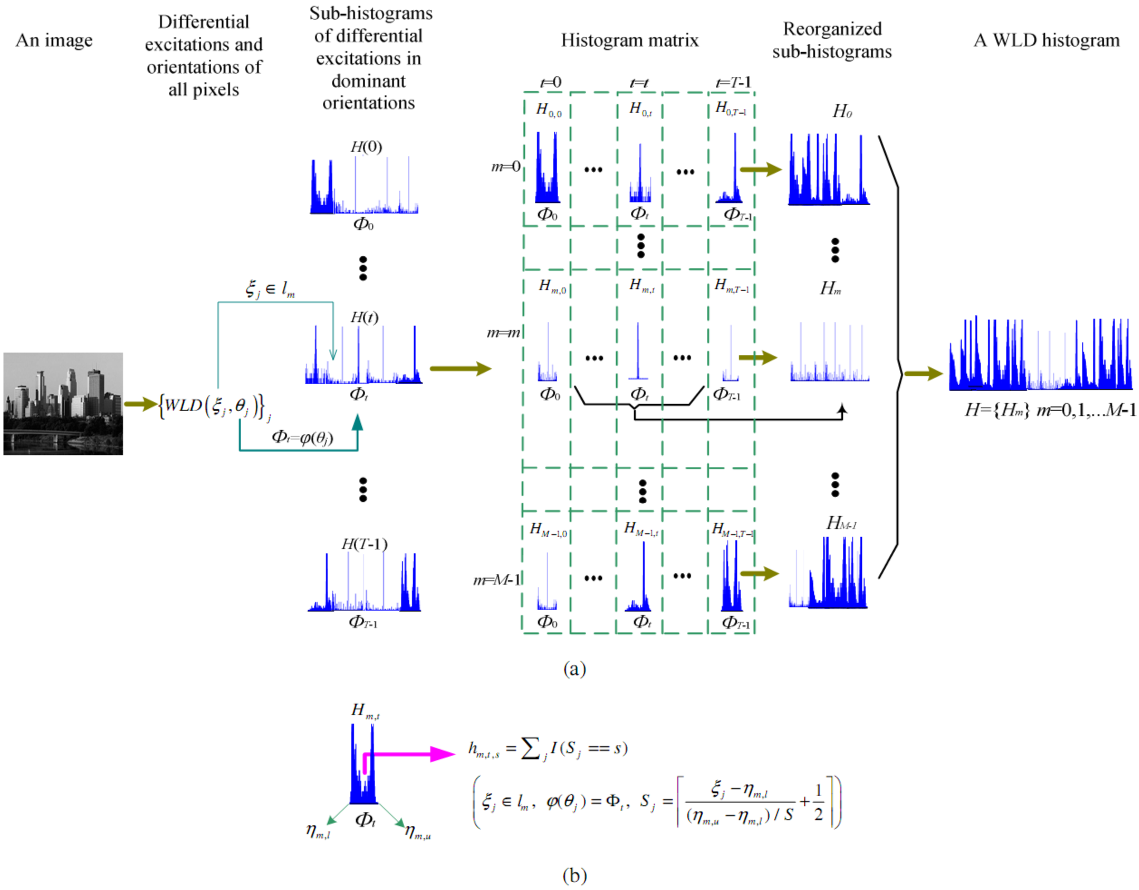

An image
Differential excitations and orientations of all pixels
Sub-histograms of differential excitations in dominant orientations
Histogram matrix
Reorganized sub-histograms
A WLD histogram
H(0)
H(t)
H(T-1)
Φ0
Φt
ΦT-1
ξj ∈ lm
{WLD(ξj, θj)}j
Φt=φ(θj)
t=0
t=t
t=T-1
m=0
m=m
m=M-1
H0
Hm
HM-1
H={Hm} m=0,1,...M-1
(a)
Hm,t
ηm,l
ηm,u
$h_{m,t,s} = \sum_j I(S_j == s)$
$\left(\xi_j \in l_m,\ \varphi(\theta_j) = \Phi_t,\ S_j = \left[\frac{\xi_j - \eta_{m,l}}{(\eta_{m,u} - \eta_{m,l})/S} + \frac{1}{2}\right]\right)$
(b)

Figure 15. An illustration of WLD histogram extraction for a given image (Chen, Shan et al. 2008).

As it is illustrated in Figure 15, WLD extracts for each pixel I_c, in a given image, two measures namely, differential excitation $\xi(I_c)$ and orientation $\theta(I_c)$. These two measures can be calculated using Eq. (15) and Eq. (16) respectively.

$$\xi(I_c) = arctan\left[\sum_{i=0}^{p-1}\left(\frac{I_i - I_c}{I_c}\right)\right] \tag{15}$$

$$\theta(I_c) = median(\frac{I_{R(i+4)} - I_i}{I_{R(i+6)} - I_{R(i+2)}}) \tag{16}$$

such that, $i\epsilon(0,1,2,3,\dots,p-1)$ are indices of I_c's neighbors, and $R(x) = mod(x,p)$,.

After having $\xi(I_c)$ and $\theta(I_c)$ extracted from all pixels of the image, WLD subsample the value space then map each differential excitation and orientation to its corresponding bin. The result is, therefore, a matrix *M* where each cell *M(i, j)* represents the occurrence frequency of a pixel that has the values i as orientation and j as differential excitation. Finally, by combining all rows of this matrix (each row represents a sub-histogram of some specific orientation) we get the final WLD histogram which can be used, later on, for matching process or similarity calculation.

II.5 Dimensionality Reduction

Sometimes, feature vector dimensions are too high to be handled in a reasonable amount of time, which make them unsuitable especially for real time systems. Selecting a feature subset from an initial bigger set of features reduces the space dimensionality by discarding the "less significant" features. These technics aim to minimise the error of approximation by projecting the original data vectors to a lower-dimensional space. The main goal of such transformation is to reach a space in where the transformed features are extremely uncorrelated, that is, the covariance matrix of the transformed data set becomes diagonal. Instance of such a techniques are the principal component analysis (PCA), Karhunen-Loeve transform (KLT) and singular value decomposition (SVD). Despite particular numerical algorithms are different, all the latter methods are equivalent in essence. The data-set scatter (i.e., variance) around each new coordinate axis decides which subset of variables should be chosen. This subset usually have to preserve a high scatter percentage (e.g., 95% or 99%) of the original dataset. By discarding dimensions with smaller variance, this approach minimises the mean squared error. In other words, the original vectors are closer, in Euclidean distance, to their decorrelated projections than other transformations. However, this approach suffers from a number of shortcomings such as data-dependency, computationally expensive, and it does not suite for dynamic data sets.

II.5.1 Principal Component Analysis

Principal component analysis (PCA) aims to draw a smaller number of uncorrelated vectors called "*principal components"* by operating a linear transformation on a big set of possibly highly correlated feature vectors. The maximal variability (i.e., scatter) in the feature vectors projected onto this the first principal component, and the second component accounts for the second maximal remaining variability, and so on. The flowing formula explains how to obtain the principal components f_k with $k = 1 \dots K$, m is the mean, and S is the covariance matrix.

$$f_k = [f_{k,1}, f_{k,2}, \dots, f_{k,n}]^T; m = \frac{1}{K}\sum_{k=1}^{K} f_k \,; S = \frac{1}{K-1}\sum_{k=1}^{K}(f_k - m)(f_k - m)^T \qquad (17)$$

S_{ij} measures the covariance between the vector components f_i and f_j, and S_{ii} is the variance of the component f_i . S is a symmetric matrix that has an orthogonal basis of eigenvectors e_i.

$$e_1, e_2, \dots, e_n \Leftrightarrow Se_i = \lambda_i e_i; i = 1,2,\dots,n; e_i^T e_j = \begin{cases} 1 & if\ i = j \\ 0 & otherwise \end{cases}; \lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_3 \qquad (18)$$

The feature variability that is concentrated along each particular eigenvector is specified by the proportion of the corresponding eigenvalue compared to the total sum of eigenvalues. In other words, eigenvectors are ordered, from the biggest to the smallest, according to the decreasing eigenvalues λ_i. Thus, the first eigenvector is directed along the line where the biggest feature scatter is projected within the new space; the second eigenvector is directed along the line corresponding to the biggest remaining scatter, and so forth (Figure 16). In preference using only a few basis vectors of the orthogonal basis instead of using all the covariance-matrix eigenvectors. Hence, PCA allows highly reducing the data dimensionality without information loss (or minimum losses), and thus simplify the representation.

$$\tilde{f}_k = \sum_{j=1}^{v} w_{k,j} e_j \,; w_{k,j} = f_k^T e_j = \sum_{i=1}^{n} f_{k,j} e_{j,i} \qquad (19)$$

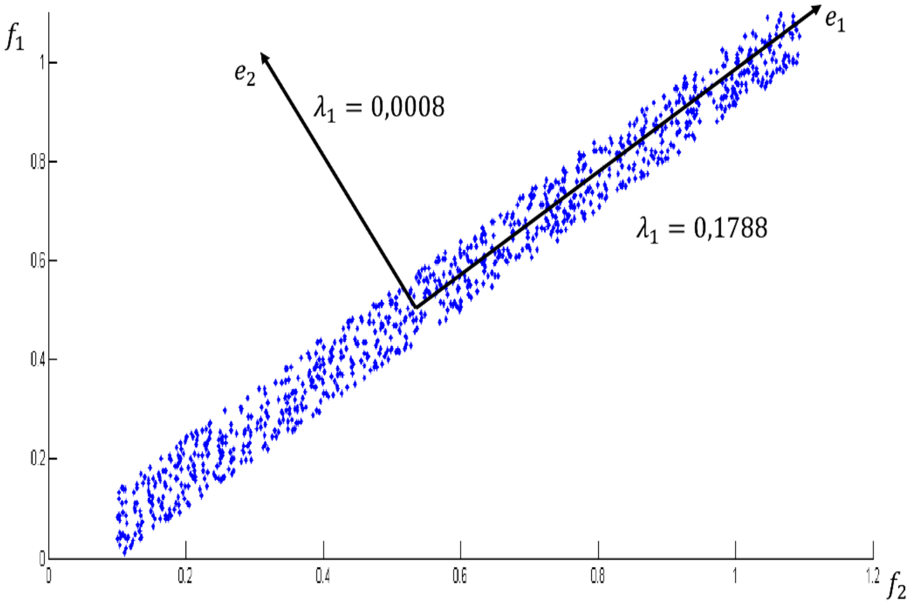


Figure 16. Dimensionality reduction

Figure 16 shows an example for a set of artificial data (depicted with small blue dots) represented in 2D space defined by the feature vectors **f** = (f_1, f_2). Applying PCA to these data results in the two eigenvectors drawn by black arrows, where the first eigenvalue $\lambda_1 = 0.1788$ is much greater than the second eigenvalue $\lambda_2 = 0.0008$. This result indicates that almost all the feature variation are held by the first eigenvector, which means that this data can approximately be represented with a 1D space obtained by the projection of the data onto the line along the first eigenvector.

II.5.2 Multidimensional Scaling

Multidimensional scaling is an alternative of PCA, which is based on a nonlinear mapping from k-dimensional feature space to n-dimensional one ($k << n$). Yet, no general theory or precise definition has been given to this approach. Although numerous statements of this approach exist, *multidimensional scaling* attempts to minimise diversities of pairwise Euclidean-distances between objects in a dataset.

Commonly, a nonlinear mapping provides better dimensionality reduction than linear methods. However, it suffers from the expense of heavier computations. In addition, this approach is data-dependent too, and it poorly suits for dynamic datasets.

II.5.3 Geometric Hashing

In contrast to PCA and multidimensional scaling, geometric hashing is a data independent approach, which performs data mapping from *n*-dimensional feature space to a very low-dimensional ones (i.e., the 1D real line or the 2D real plane). Generally, it uniformly spreads the given data over the range of a low-dimensional space, such that the metric properties of the hashed space significantly differ from those of the original space. In case when only the local metric properties of a low-dimensional feature space are needed to be maintained, geometric hashing is therefore applied for indexing. However, dimensionality increasing of the original space leads to difficulties in designing a good hashing function.

Generally, dimensionality reduction facilitates the indexing of multidimensional feature spaces, and the data analysis is then performed on the transformed data rather than the original one. It also eliminates the impacts of the "curse of dimensionality" issue and improves the results.

II.6 Conclusion

In this chapter, we have seen how an image can be represented using a derived values, that are called features, from its content. These features were mainly categorized into four main approaches namely color, shape, interest-point and texture features. There is no generic feature that can be used for all kind of image analysis tasks, instead, different features could be combined (e.g., color and texture) for better image representations. Moreover, we found out that most of the features such as GLCM suffers from numerous drawbacks and needed to be improved or generalized to cover more representation aspects. In this latter chapter, we have seen how dimensionality reduction of feature vectors operates. This latter facilitates the indexing of multidimensional feature spaces and eliminates the impacts of the "curse of dimensionality" issue.

Chapter III. GLCM GENERALIZATION, RELATED WORK

Co-occurrence matrix-based features describes the interaction between pixels of a given neighbourhood. These features interest in extracting statistics that describe this interaction in effective way (i.e., correlation, variance, etc.). Many Co-occurrence matrix-based features have been proposed in literature, the GLCM is amongst the widely known ones. In this chapter, we list the most relevant literature works which are based on extracting color (resp., gray-level) co-occurrence matrix from images.

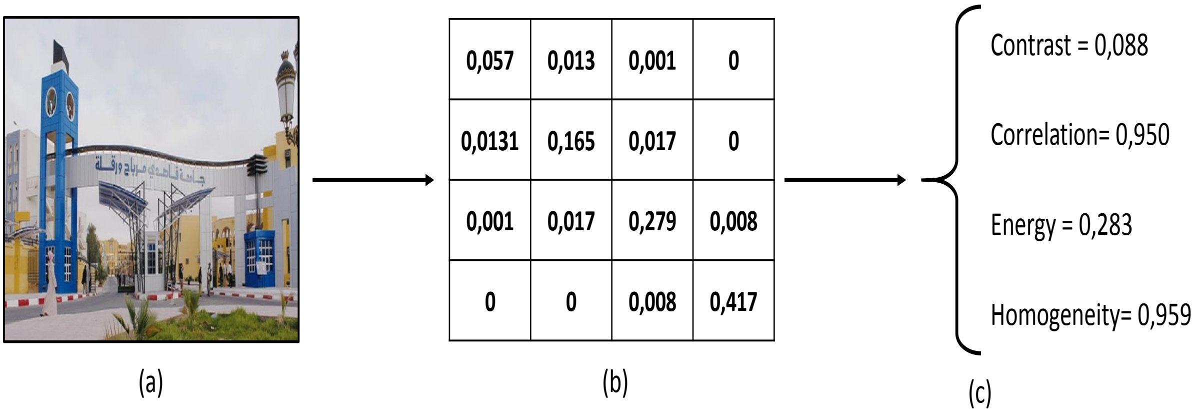
III.1 Gray Level Co-occurrence Matrix

The GLCM(Haralick and Shanmugam 1973) is generated by calculating the occurrence frequency of gray-level pairs within a texture image. In simpler words, let us suppose that M is a GLCM. Then, each element $M(i, j)$ represents the frequency of a gray-level j that appears at a given offset $(\Delta x, \Delta y)$ from a gray level i. For a given image I, and for an offset$(\Delta x, \Delta y)$, the GLCM can be extracted using Eq. (20).

$$GLCM(i,j) = \sum_{p=1}^{N}\sum_{q=1}^{M}\begin{cases}1, & if\ I(p,q) = i\ and\ I(p+\Delta x, q+\Delta y) = j\\ 0, & otherwise\end{cases} \tag{20}$$

After having proposed the GLCM, Haralik has extracted a set of features from it, namely, angular second moment, contrast, correlation, variance, inverse different moment, sum average, sum variance, sum entropy, entropy, difference variance, difference entropy, maximal correlation cofficient and three other mesures of correlation namely *f11*, *f12* and *HXY*.

Figure 17 shows an exemple of extracting the co-occurrence matrix from a given image, and then extracting some features from this matrix.



0,057	0,013	0,001	0
0,0131	0,165	0,017	0
0,001	0,017	0,279	0,008
0	0	0,008	0,417

Contrast = 0,088

Correlation= 0,950

Energy = 0,283

Homogeneity= 0,959

Figure 17. Extracting GLCM features from an image. (a) is a given image, (b) is the extracted GLCM (*M*) and (c) represents four statistical features extracted from this GLCM.

In addition to its simplicity, the GLCM has proven its ability to represent the gray-level texture in a good way. However, since it holds information about gray levels only, the GLCM is not able to deal with color textures. Integrating color information into the GLCM is still an open issue.

Although it is a fourty-years old method, many comparative studies have proven that GLCM is one of the best texture representations. We mention the work of Ohanian and Dubes(Ohanian and Dubes 1992) , where they compare and evaluate four texture methods, namely GLCM(Haralick and Shanmugam 1973), Markov Random Field(Cross and Jain 1983), fractal-based method(Pentland 1984) and multi-channel filtering(Farrokhnia and Jain 1991).

Their evaluation involves four test sets, two of them are synthetic and the others are natural. Results indicate that the GLCM outperforms the other methods. Nurhaida et al.(Nurhaida, Manurung et al. 2012) have compared the performance of three methods for the classification of Batik texture dataset. These methods are GLCM, Canny Edge Detection, and Gabor Filters. Experimental results showed that the GLCM has again yielded the best classification accuracy.

Although the GLCM has proven its effectiveness as a texture representation, four main issues should be addressed in order to improve its accuracy and the results it yields:

Color integration: Since the GLCM is computed from gray-level images only, color information, which may be very important, is totally ignored. Several studies have been conducted to tackle this issue(Palm 2004, Arvis, Debain et al. 2011)

Multi-spectral integration: some attempts have been done to integrate information extracted from the different spectral bands, composing a multi-spectral image, into a co-occurrence matrix(Khelifi, Adel et al. 2010).

Color sampling and mapping: sampling is the process of dividing a space into a set of bins, whereas, mapping is the process of determining the bin that corresponds to a given value. Color sampling and mapping are critical processes since the better they are, the better results are. Vadivel et al.(Vadivel, Sural et al. 2005) propose a mapping method where they take into account information about both color and gray level.

Extracting new features: Besides the features introduced by Haralick in his original work on GLCM(Haralick and Shanmugam 1973), many other authors have proposed new ones. Examples include(Dacheng, Xuelong et al. 2002, Palm 2004).

This thesis addresses two issues among those cited above. The first one concerns color integration while the second is sampling and mapping.

III.2 Combining and Integrating Color

As we have explained before, GLCM is a representation of a texture image in the gray-level space (i.e., mono channel). Color integration is the process of generalizing the GLCM in order to make it able to represent color images (i.e., multi-channel). To fulfill such a task, a number of color spaces have been used in literature, such as Red, Green and Blue (RGB); HSV; Luminance, a-axes and b-axes (L*a*b); Luv; and Hue-Max-Min-Diff (HMMD)(Stanchev, Green Jr et al. 2003).

III.2.1 Combining GLCM with HSV Color Histogram

In order to improve CBIR accuracy and efficiency, authors of (Kong 2009, Reddy 2010) suggest to use both color and texture features as image descriptors. In order to separate chromatic from the achromatic colors, all features has been extracted within HSV color space. Firstly, he has quantized hue channel into eight bins according to the color model of substantial analysis. Both saturation and intensity were quantized into three bins in accordance with the human vision thresholding system. Then, Kong extracted, as color feature, a $3 \times 3 \times 8 = 72$ length cumulative color histogram as the next formula shows.

$$Hist(k) = \sum_{i=1}^{N}\sum_{j=1}^{M} s\big(I(i,j)\big); s(p) = \begin{cases} 1 \; if \; 9H_p + 3S_p + V_p = k \\ 0 \qquad\qquad otherwise \end{cases} \tag{21}$$

where I is an image that has height N and width M. H_p, $S_p$and V_p respectively are the bins in which hue, saturation and intensity of the corresponding pixel p are falling in.

The adopted channel quantization is presented in the following formulas. Where H, S, and V respectively are the mapping results for hue *(h)*, saturation *(s)* and intensity *(v)* channel for a given pixel p.

$$H = \begin{cases} 0 & if \; h \, \epsilon[316, 20] \\ 1 & if \; h \, \epsilon[31, 40] \\ 2 & if \; h \, \epsilon[41, 75] \\ 3 & if \; h \, \epsilon[76, 155] \\ 4 & if \; h \, \epsilon[156, 190] \\ 5 & if \; h \, \epsilon[192, 270] \\ 6 & if \; h \, \epsilon[271, 295] \\ 7 & if \; h \, \epsilon[296, 315] \end{cases} \tag{22}$$

$$S = \begin{cases} 0 & if \; s \, \epsilon[0, 0.2[\\ 1 & if \; s \, \epsilon[0.2, 0.7[\\ 2 & if \; s \, \epsilon[0.7, 1] \end{cases} \tag{23}$$

$$V = \begin{cases} 0 & if \; v \, \epsilon[0, 0.2[\\ 1 & if \; v \, \epsilon[0.2, 0.7[\\ 2 & if \; v \, \epsilon[0.7, 1] \end{cases} \tag{24}$$

Such a quantification is effective in minimizing effects of light intensity, and reducing the computational complexity and therefore the execution time.

After having the cumulative color histogram extracted, they combined it with four of GLCM features namely, energy, contrast, entropy and inverse difference. The proposed approach has been evaluated using a CBIR system. Outcomes reported in Kong's experiments indicate that after combining GLCM with the cumulative color histogram the results were well improved.

Later on, the same approach has been adopted by (Kavitha, Rao et al. 2011). This time however, each image is explicitly divided into a set of six equal sized sub-blocks, and then extract from each sub-block a cumulative histogram in addition to four of GLCM features namely, energy, contrast, entropy and inverse difference. Finally, all the extracted features are combined in order to formulate the final feature vector. According to their experiments, the proposed approach outperforms the former one in which sub-dividing the image is not considered.

However, such approaches neglect the correlation between the color and gray-level features. Additionally, weights for color feature and texture features must be explicitly defined which raises another problem of weight assignment.

III.2.2 Combining GLCM with RGB Color Moments

Another attempt to resolve the lack of color information in case of using GLCM has been done by (Huang 2007, Kaya and Kayci 2014). In the latter work, an application of neural network and image processing techniques has been presented for detecting and classifying a set of the pre-mentioned Phalaenopsis seedling diseases. The author decided to use nine texture features extracted from GLCM along with other three color-features extracted using RGB color space. The texture features are contrast, uniformity, maximum probability, homogeneity, inverse difference, difference variance, diagonal variance, entropy, and difference entropy, whereas the color features are the red channel mean (R_{mean}), green channel mean (G_{mean}), and blue channel mean (B_{mean}).

As a classifier, Huang used a back propagation neural network (BPNN) which consists of three layers: input layer, hidden layer, and output layer. The input layer has 21 nodes, three of them are for color features (*Rmean*, *Gmean* and *Bmean*), while the remaining 18 ones are for texture features, as Figure 18 shows. The output layer was made of four nodes correspond to four categories namely, BSR, PBR, BBS and OK. At the beginning, the number of nodes n_h in the hidden layer was calculated using formula in Eq. (25.

$$n_h = \left[\frac{n_i + n_o}{2}\right] + \sqrt{n_p} \tag{25}$$

such that n_i and $n_o$are the number of input and output nodes respectively, and n_p indicates the number of images in the training set.

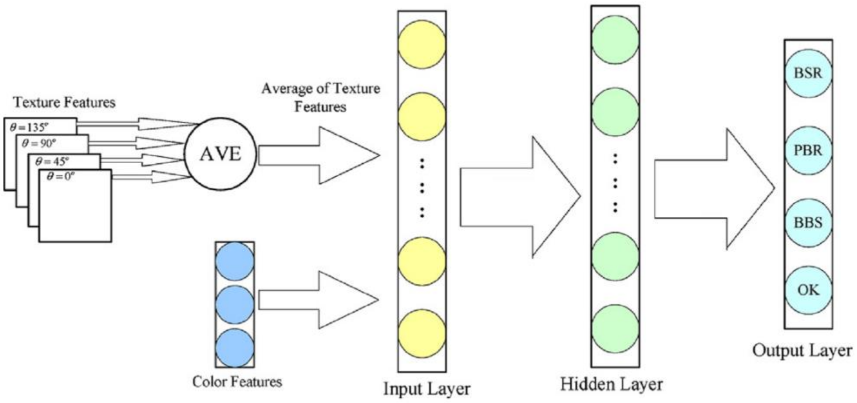


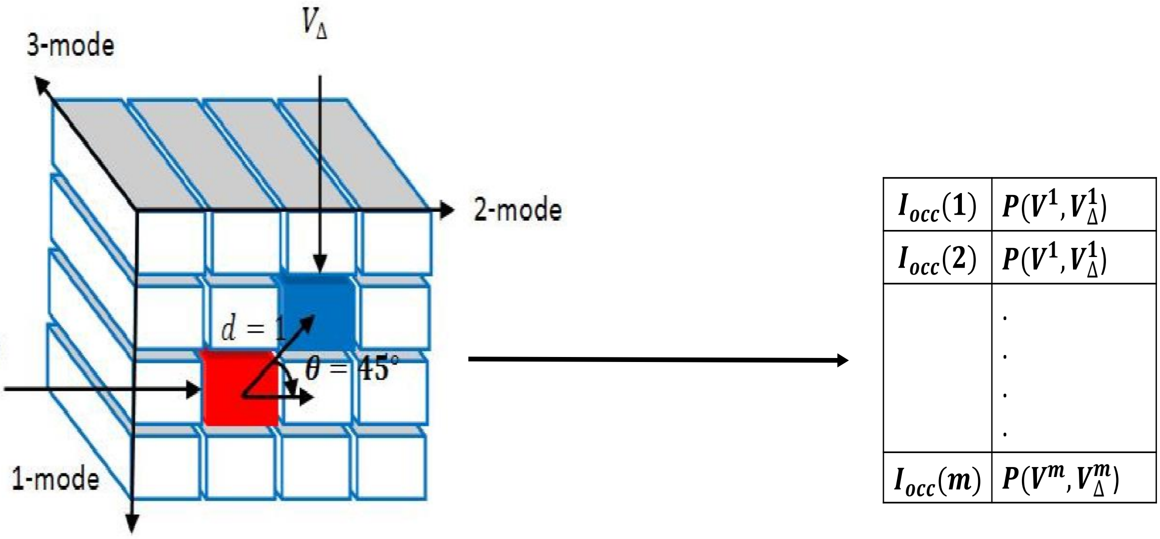
Figure 18. General shceme of the used BPNN classifier(Huang 2007).

Likewise, the same features (i.e., dominant color descriptor (DCD)(Chang, Sikora et al. 2001) and GLCM) in these works has been adopted by(Rao, Rao et al. 2011). After dividing the image into 8-coarse partitions, Rao et al. extracted those features along with invariant moments of gradient vector flow fields, as shape features, from each partition. Finally, they construct a combined feature vector for color, texture and shape. This final feature vector has been used to evaluate e CBIR system where the result indicates a slightly improvement.

As features are extracted separately from each channel of the color space, these approaches suffer from the absence of correlation between color-texture features and color-color features.

III.2.3 Generalized Gray-Level Difference Method

In(Khelifi, Adel et al. 2010), the authors propose to use a Generalized Gray-Level Difference Method (GGLDM) as an extension of the GLCM to multi-spectral images by assuming joint information between different spectral bands. The co-occurrence probabilities are not calculated between gray levels, but rather between vectors of gray levels. Each component in a vector represents a value of a given pixel within a specific spectral band. The distance between vectors is calculated using the probability function P(G(V; VΔ|d; θ) = (k1; k2:::kNb)), where V and VΔ are two vectors with spectral components (i1; i2; i3:::iNb) and (j1, j2; j3:::jNb), respectively, d is a given distance and θ is a given direction (Figure 19).



$I_{occ}(1)$	$P(V^1, V_\Delta^1)$
$I_{occ}(2)$	$P(V^1, V_\Delta^1)$
	. . . .
$I_{occ}(m)$	$P(V^m, V_\Delta^m)$

Figure 19. Extraction GGLDM from a multi-spectral image.

Subsequently, authors have modified the original GLCM's features to fit their new representation as Eq. (26)-(30) show:

$$\text{Contrast} = \sum_{i=1}^{i=m} \left\|D^i\right\|^2 P(V^i, V_\Delta^i) \tag{26}$$

$$\text{Angular Second Moment} = \sum_{i=1}^{i=m} P(V^i, V_\Delta^i)^2 \tag{27}$$

$$\text{Entropy} = \sum_{i=1}^{i=m} P\left(V^i, V_\Delta^i\right) \log\left(P\left(V^i, V_\Delta^i\right)\right). \tag{28}$$

$$\text{Mean} = \sum_{i=1}^{i=m} \left\|D^i\right\| P(V^i, V_\Delta^i) \tag{29}$$

$$\text{Inverse Difference Moments} = \sum_{i=1}^{i=m} \frac{P(V^i, V_\Delta^i)}{\left\|D^i\right\|+1}. \tag{30}$$

where $\left\|D^i\right\|^2 = \left[(k_1^i)^2 + (k_2^i)^2 + \cdots + (k_{Nb}^i)^2\right]$, N_b is the number of columns in *GGLDM*.

Similarly, other literature works tried to generalize the co-occurrence matrix to multi spectral images. In (Hauta-Kasari, Parkkinen et al. 1996), authors define a set of color classes for a pixel to be a part of. After mapping each color spectrum in the multi-spectral image to its appropriate class, they extract their final co-occurrence matrix. In (Huang, Liu et al. 2014), authors proposed a representation of texture in multi/hyper spectral images using clustering algorithms (i.e., k-means and fuzzy c-means) and sparse representation.

Although these methods can be used to represent images in different color spaces (e.g., RGB, HSV or L*a*b*) by considering each channel as a spectral band, they does not take into account the correlation between these channels. In addition, they ignores the fact that some spectral bands are more important than others are.

III.2.4 HSI Co-occurrence Matrix

In(Shearer and Holmes 1990), the authors have proposed to use Hue, Saturation and Intensity (HSI) color space to derive, from the image, three co-occurrence matrices that attribute hue, saturation and intensity. Then, they have extracted a set of 33 features (i.e., 11 features per matrix) from these matrices in order to be used in texture image discrimination. However, the major weakness of this method is that it ignores the correlation between the HSI color channels.

Many other works in literature have adopted the HSI co-occurrence matrix. In (Shuttleworth, Todman et al. 2002), the authors adopted this method in order to automatically classify images of colon tissue and identify the colorectal cancer. Results showed that this method outperforms the GLCM. Similarly, authors of (Pydipati, Burks et al. 2006) suggested using this method to identify diseased and normal citrus leaves under laboratory conditions. Experimental results showed a high classification accuracy for all leaf classes. In (Drimbarean and Whelan 2001), the authors conduct a comparative study between different texture features namely Distance Cosine Transform (DCT)(Unser 1986), Gabor filter-based features(Daugman 1980, Daugman 1985) and a co-occurrence matrix extracted using HSI color space. The obtained experimental results indicate that incorporating color information improves the performance of the examined texture analysis techniques.

III.2.5 HSI-Luminance Co-occurrence Matrix

In addition to the 33 features proposed by Shearer(Shearer and Holmes 1990), Chang et al.(Chang, Zaman et al. 2012) have included 11 other features that have been extracted from a co-occurrence matrix of National Television System Committee (NTSC) luminance (L). Nevertheless, this method suffers from the same drawback which is the absence of correlation between the color channels.

III.2.6 Hue Co-occurrence Matrix

Shim et al.(Shim and Choi 2003) tried to resolve the issue of integrating color into the GLCM by extracting it from the hue channel of the HSV color space. In their study, they have considered quantizing the color space into 16 bins, in addition to a displacement vector that is defined as: $(\Delta x, \Delta y)\ \epsilon\ ((1,0),(-1,0),(0,1),(0,-1))$ (i.e., only four adjacent neighbours) which was sufficient for their experiments.

In simpler words, let I be an $N \times M$ image quantized into m colors, and *p(x, y)* is the color value of a pixel with the index *(x, y)*. Then, Shim et al.'s simplified CCM is given by Eq. (31).

$$H^1(i,j) = \eta(\left(p(x,y), p\left(N_{(x,y)}\right) == (i,j)\right) = \alpha \sum\nolimits_{(x',y')\in N_{(x;y)}} C_j(x',y') \qquad (31)$$

where η indicates the number of times that the pixel pair $(p(x,y), p(N_{(x,y)}))$ equals the value of the color indices (i,j), $N_{(x,y)}$ refers to the 4-neighbors of the pixel $p(x,y)$, α is a normalization constant which equals $(N \times M)/4$, and $C_j(x,y)$ is given by Eq. (32).

$$C_j(x,y) = \begin{cases} 1 & if\ p(x,y) = i \\ 0 & else \end{cases} \tag{32}$$

Shim et al. does not derive any statistical moments from the proposed Color Co-occurrence Matrix (CCM). Instead, they concatenated its diagonal and non-diagonal elements in order to formulate a final histogram which is used to index texture images. The latter step was justified by the claim that homogeneous color regions of the image contribute to the diagonal elements of CCM whereas the non-homogeneous regions to the non-diagonal elements of CCM. The feature vector of an image I is given, therefore, by Eq. (33).

$$F^1 = (H_D^1, H_N^1) \tag{33}$$

where, $F^1$is the Modified CCM (MCCM), H_D^1 and H_N^1 are, respectively, the diagonal and non-diagonal elements of CCM.

Despite it incorporates color information, CCM suffers from numerous drawbacks. First, by limiting themselves to color only (i.e., hue channel), information about intensity is completely lost. Which means that it is able to deal only with mages that are represented in single channel space. Second, using the CCM's diagonal and non-diagonal elements, as they are, instead of deriving third order moments from them, makes it very sensitive to color changes and cost sometimes more computation. In simpler words, two images that represent the same texture but in different colors cannot be yielded as similar by CCM.

III.2.7 Single-Channel & Multi-channel Co-occurrence Matrices

Another attempt to integrate color has been done by Palm(Palm 2004). He has proposed two alternative methods using Luv color space. He justified using Luv instead of RGB by the fact that RGB color space channels are highly correlated (Tan and Kittler 1994, Wandell 1996). Such a correlation between the channels makes the features very sensitive to tinting, toning and shading. Thus, he has transferred images from RGB into Luv color space, where the channels are de-correlated, using the formula in Eq. (34).

$$\begin{pmatrix} L \\ U \\ V \end{pmatrix} = H_{LUV} \cdot \begin{pmatrix} R \\ G \\ B \end{pmatrix} = \frac{1}{\sqrt{6}} \begin{pmatrix} \sqrt{2} & \sqrt{2} & \sqrt{2} \\ 2 & -1 & -1 \\ 0 & \sqrt{3} & -\sqrt{3} \end{pmatrix} \begin{pmatrix} R \\ G \\ B \end{pmatrix} \tag{34}$$

After transferring the image into Luv color space, Palm suggested using two methods for extracting the co-occurrence matrix. In the first method, three Single-channel Co-occurrence Matrices (SCM) has been defined. He firstly assumed a limited k-dimensional Cartesian space like RGB or LUV with k= 3. Then, each SCM (i.e., SC_d^k) corresponds to a GLCM that is extracted from the kth color channel (i.e., f_k) of the image, where $f = (f_1, f_2, \dots, f_k)^T$ is calculated using Eq. (35).

$$\Pr\left(w^k, \overline{w}^k\right) = SC_d^k(w, \overline{w}) = \Pr(f_k(p_1) = w \wedge f_k(p_2) = \overline{w} \wedge |p_1 - p_2| = d) \tag{35}$$

where $\Pr\left(w^k, \overline{w}^k\right)$is the joint probability that indicates the adjacency of w^k and $\overline{w}^k$ on the same channel f_k; $p_1 and\ p_2$are two pixels from the image, and d is a displacement distance.

The major weakness of SCM is that it completely ignores the correlation between the three channels. In addition, it considers all channels as important as the same, however, some channels are less important comparing to others (e.g., image shading is not as important as color changing).

In the second method, the correlation was partially covered by using three Multi-channel Co-occurrence Matrices (MCM). Two of these matrices, namely LV and UV represent the chromatic information, whereas, the achromatic information is represented by the LU matrix. In other words, Palm introduces multi-channel co-occurrence matrices (MCMs) $MC_d^{k1,k2}$ which count the pairwise occurrence of values in two different bands f_{k1} and f_{k2}, as Eq. (36) shows.

$$\Pr\left(w^{k1}, \overline{w}^{k2}\right) = MC_d^{k1,k2}(w, \overline{w}) = \Pr(f_{k1}(p_1) = w \wedge f_{k2}(p_2) = \overline{w} \wedge |p_1 - p_2| = d) \tag{36}$$

where $\Pr\left(w^{k1}, \overline{w}^{k2}\right)$is the joint probability that indicates the adjacency of $w^{k1}$and $\overline{w}^{k2}$ on the channels f_{k1} and f_{k2}, respectively. p_1 and p_2 are two pixels from the image, and d is a displacement distance.

By adopting this latter method, true, tint, tone and shaded colors are taken into account. However, it neglects the correlation between the chromatic and the achromatic channels.

III.2.8 Color-features combined GLCM

Arvis et al.(Arvis, Debain et al. 2011) tried to generalize the GLCM to multispectral images. Let $C_1, C_2, \dots C_n$be the channels of color space in which the image has been represented. Thier generalization can be calculated according to Eq. (37).

$$M_{d(C_u,C_v)}(i,j) = \#\left((s, s+d) \in R^2 \middle| C_u(s) = i \ \wedge C_v(s+d) = j\right) \qquad (37)$$

where s is a given point within the channel C_u of the image, and d is a displacement vector.

In their method, colour images are coded using three channels, leading to six different matrices. The first three matrices, namely (R,R), (G,G), (B,B), represents the gray-level information and can be extracted from each single channel as standard greyscale co-occurrence matrices. Whereas, the three others, namely (R,G), (R,B), (G,B), represents the color information and take into account the correlations between two channels each time. Therefore, this method lead to a total of 30 texture features (5 features from each matrix). Figure 20 illustrates the scheme corresponding to this method.

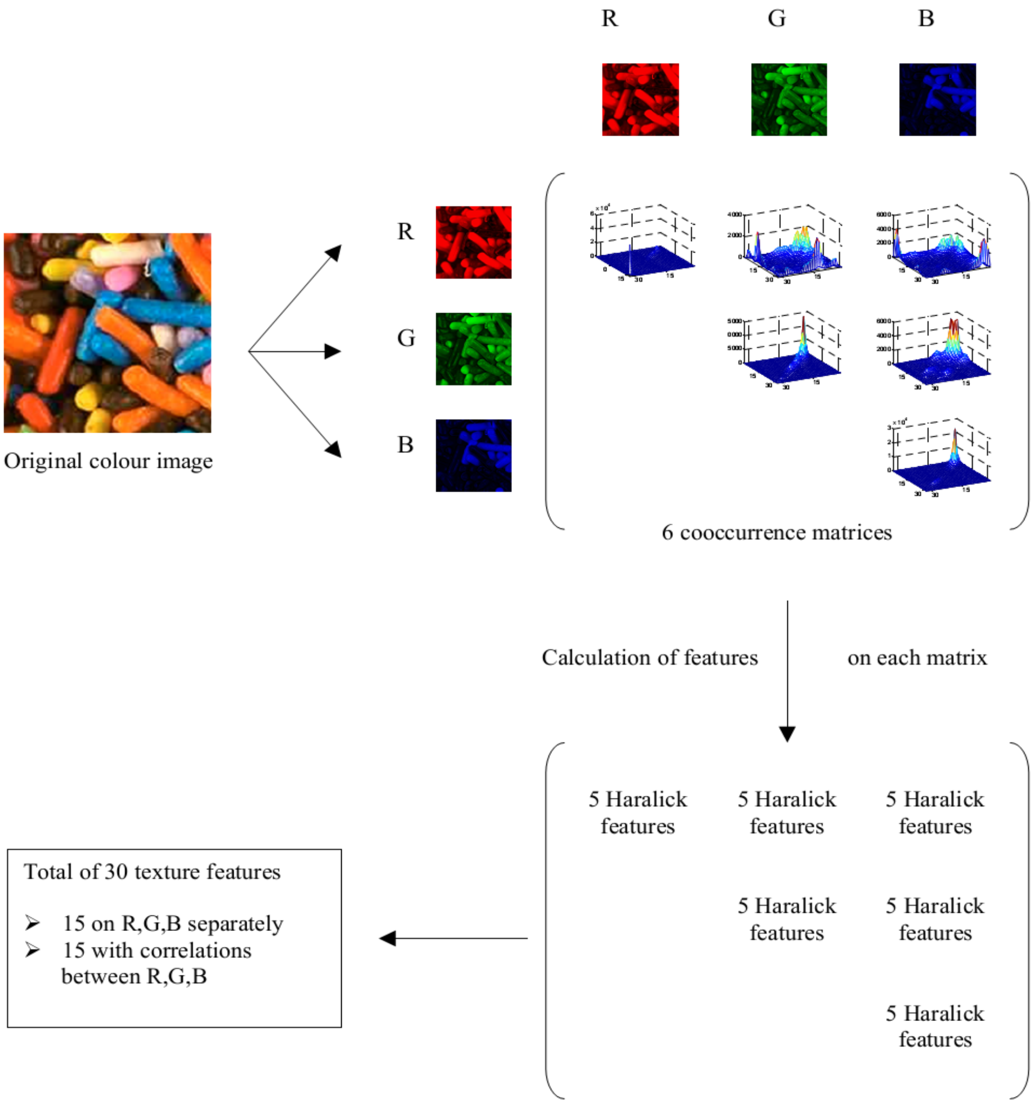


Figure 20. Illustration of the multispectral RGB-method applied to an image from the class FOOD0006 from VisTex dataset(Arvis, Debain et al. 2011).

An alternative solution proposed by Arvis et al. to reduce the number of features. It consists in changing the color space of the image in order to obtain one channel for luminance and two others for chrominance information, as Figure 21 shows. Therefore, they use HSV color space because it corresponds better to how people experience colour than the RGB does.

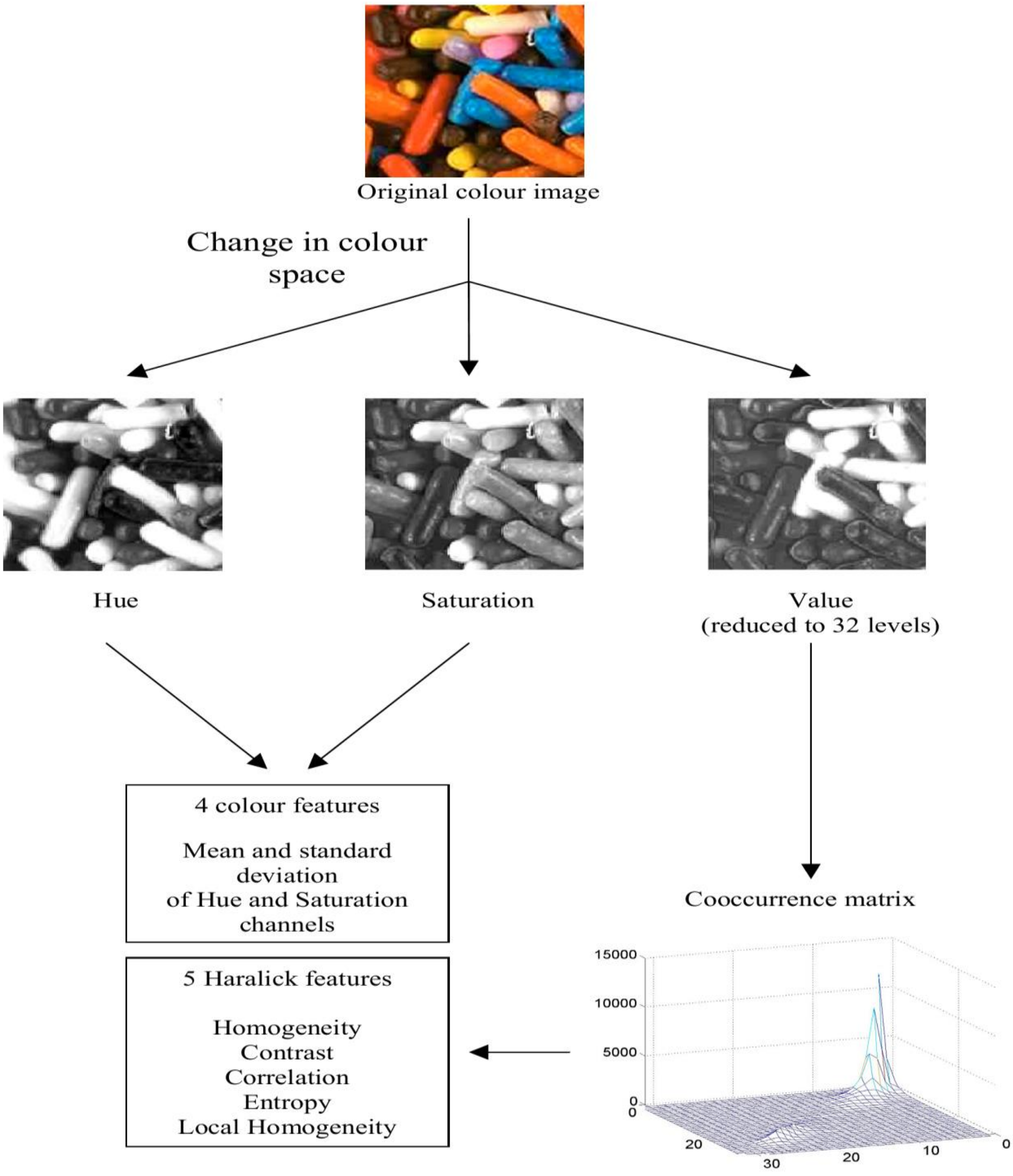


Figure 21. Illustration of the multispectral HSV-method applied to an image from the class FOOD0006 from VisTex(Arvis, Debain et al. 2011).

In the later method, the correlation between the features extracted from *Value-channel* matrix and those of *Hue* and *Saturation channels* is not covered. Another issue this method suffers from is how to choose suitable weights to be assigned to each of the GLCM and color features.

III.3 Color Sampling and Mapping

Human vision is a quite complicated process. At low illumination, only the rod cells are stimulated, which leads to gray-level perception of objects. However, as illumination increases, the cone cells become more stimulated, which leads to color perception(Vadivel, Sural et al. 2005). Hence, various color spaces, including HSV, Luv, L*a*b and XYZ have been introduced in order to simulate this complicated process of gray-level/color perception. After having the color space chosen, a sampling process is need to be set. The sampling is the process of dividing the channel into a set of ranges. Whereas the mapping is process that defines for a given value the range it belongs to.

III.3.1 Integrated Color Intensity Co-occurrence Matrix

In order to differentiate scotopic from photopic vision, one may think to establish a threshold to distinguish a low luminance from a high one. However, such a hard thresholding could result in a confusion near the boundaries between high and low illuminations. Vadivel et al.(Vadivel, Sural et al. 2005) attempted to resolve this issue by trying to understand how human vision operates and by analysing the visual properties of the HSI color space. Then, they employed saturation and intensity channels to determine the extent to which a given pixel is a color W_{col} and the extent to which it is a gray level W_{gray}. Given a pixel p with three components *(H, S, I)* which corresponds to *hue*, *saturation* and *intensity* respectively, then, W_{col} must satisfy the following constraints:

a) $W_{col}(S, I) \in [0,1]$.

b) $If\ S_1 > S_2$ then $W_{col}(S_1, I) > W_{col}(S_2, I)$.

c) $If\ I_1 > I_2$ then $W_{col}(S, I_1) > W_{col}(S, I_2)$.

d) $W_{col}(S, I)$ changes slowly with S for high values of *I*.

e) $W_{col}(S, I)$ changes sharply with S for low values of *I*.

They proposed the formula given in Eq. (38) and Eq. (39) in order to calculate the extent of color and gray level, respectively, where S stands for saturation , I for intensity, r_1 and

r_2 are constants that depend on the particular application in which ICICM is used. In(Vadivel, Sural et al. 2005) $r_1$and r_2 have been estimated empirically.

$$\mathrm{W_{col}(S,I)} = \begin{cases} S^{r_1*(\frac{255}{I})^{r_2}} & for\ I \neq 0 \\ 0 & for\ I = 0 \end{cases} \tag{38}$$

$$W_{gray}(S,I) = 1 - W_{col}(S,I) \tag{39}$$

In order that $\mathrm{W_{col}(S,I)}$ of Eq. (38) satisfies the conditions (a)–(e), $r_1$should take a value slightly higher than 0 and r_2 should take a value slightly less than 1. Vadivel et al. have performed a set of experiments on the MIT texture dataset. They, have used different combinations of r_1 and r_2 and cal-culated recall and precision for different queries. It is found that a combination of $r_1 = 0.1$ and $r_2 = 0.85$ gives the best average recall and precision.

They applied their method to extract a color histogram that they evaluated in image retrieval context. Later on, they tried to adopt the same method for texture representation (Vadivel, Sural et al. 2007) and proposed an Integrated Color Intensity Co-occurrence Matrix (ICICM). This matrix holds both color and intensity information of a texture image. It is constituted of four sub-matrices as follows:

$$ICICM = \begin{pmatrix} ICICM_{cc} & ICICM_{cg} \\ ICICM_{gc} & ICICM_{gg} \end{pmatrix}$$

where $\mathrm{ICICM_{cc}}$ holds information about the co-occurrence of color-color pairs, $\mathrm{ICICM_{cg}}$ holds information about the co-occurrence of color-gray pairs, $\mathrm{ICICM_{gc}}$ holds information about the co-occurrence of gray-color pairs and $\mathrm{ICICM_{gg}}$ holds information about the co-occurrence of gray-gray pairs.

These matrices are updated according to Algorithm 1, where *QH* and *QI* are the quantization factors for hue and intensity, respectively. For a given pixel *px* with an index *(m, n)*, $hl_{(m,n)}$ and $gl_{(m,n)}$ denote, respectively, its hue and gray-level values, *S* is its saturation and *I* is its intensity, $hl_{N(m,n)}$ and $g_{lN(m,n)}$ are, respectively, hue and gray-level values of *px*'s neighbor (with the index *N(m, n)*), the operator $+=$ stands for incrementing variables.

Algorithm 1: Updating ICICM

```
Input  :        An image $I$
Output:         The ICICM matrix

foreach co-occurrence of pixels $(I_{(m,n)}, I_{N(m,n)})$ in $I$
    $ICICM\left[\frac{hl_{(m,n)}}{QH}, \frac{hl_{N(m,n)}}{QH}\right] += W_{col}^{(m,n)}(S,I) + W_{col}^{N(m,n)}(S,I)$
    $ICICM\left[\frac{hl_{(m,n)}}{QH}, \frac{gl_{N(m,n)}}{QI}\right] += W_{col}^{(m,n)}(S,I) + W_{gray}^{N(m,n)}(S,I)$
    $ICICM\left[\frac{gl_{(m,n)}}{QI}, \frac{hl_{N(m,n)}}{QH}\right] += W_{gray}^{(m,n)}(S,I) + W_{col}^{N(m,n)}(S,I)$
    $ICICM\left[\frac{gl_{(m,n)}}{QI}, \frac{gl_{N(m,n)}}{QI}\right] += W_{gray}^{(m,n)}(S,I) + W_{gray}^{N(m,n)}(S,I)$
end foreach
```

III.3.2 Fuzzy Color Co-occurrence Matrix

Another attempt to improve the mapping task has been carried out by Ledoux et al(Ledoux, Losson et al. 2015). They proposed a Fuzzy Color Co-occurrence Matrix (FCCM) that describes the interaction between neighboring pixels in RGB color space. For a given image I in the RGB color space, the FCCM can be calculated using Eq. (40).

$$FCCM(c, c') = \sum_{p \in I} \sum_{p' \in N_p} \min(\mu_{Sc}(p),\ \mu_{Sc'}(p')\ ) \tag{40}$$

where p is a pixel from I that has a pixel neighborhood N_p and p' is thus a neighbor of p. $c\ and\ c'$ are two bins resulted from the sampling of the color space. $\mu_{Sc}(p)$ represents the membership degree of the pixel p to the bin c. In order to calculate this degree of relationship, Ledoux et al. suggest to use either a Gaussian function (Eq. (41)) or a Triangular one (Eq. (42)).

$$\mu_{Sc}(\mathrm{p}) = \mathrm{e}^{\frac{-\|-x-c\|^2}{2\alpha^2}} \tag{41}$$

$$\mu_{Sc}(p) = \max(0, 1 - \frac{\|x-c\|}{\beta}) \tag{42}$$

where $\|x-c\|$ refers to the Euclidean distance between color x and bin c. α and β are two parameters designed to ensure that $\mu_{Sc}(p)$ converges to the value 0.5 near the boundaries of the bin c.

Regardless their good work trying to integrate the fuzziness into CCM, the method of Ledoux et al. suffers from numerous drawbacks. Firstly, by using only chromatic channels (i.e., R, G and B), information that concerns achromatic channel is totally lost. Secondly, they considered each pair of neighboring sites (p, p') contributes to $FCCM\,(c, c')$ by adding only the minimum between $\mu_{Sc}(p)$ and $\mu_{Sc'}(p')$, whereas, other non-minimum values are neglected regardless their meaning (e.g., min (1, 5) = min (1, 2)). In simpler words, a set of different pixel pairs could contribute to the same FCCM cell because they have the same minimum. Moreover, Euclidean distance is not suitable for calculating similarities between RGB colors. For example, if we have two green colors with (R, G, B) values of $c_1 = (160,190,0)$, $c_2 = (0,255,0)$ and a yellow color with an (R, G, B) value of $c_3 = (255,220,70)$, then, an euclidean distance between these three colors tells us that c_1 is more similar to c_3 than $c2$ ($\|c_1 - c_3\| = 134, \|c_1 - c_2\| = 172$), which is incorrect.

III.4 Conclusion

In this chapter, we have seen that many works attempted to improve the GLCM. The main issue these works focused on is how to incorporate color information within GLCM without resorting to any other kind of features. They confronted many difficulties trying to deal with presence of multitude of channels. Other problem is how to perform an optimal color sampling and mapping task, and avoid the problems the hard mapping raises. It is true that the proposed improvements have enhanced the results. However, it incorporates new issues as loss of correlation between different color channels, the ignorance of some color channels, the high computation, etc.

Chapter IV. CONTENT -BASED IMAGE RETRIEVAL

Now days, large amounts of internal and external memory become increasingly cheaper. Hence, both public and personal image datasets have grown so rapidly. Image datasets are meant to store different kinds of art collections as, medical, satellite, and general photograph collections. These datasets might be used for various applications. For example, User may want to find images of *'elephants'*, another may want *'sunrise'* or may even abstract concepts such as *'happiness'* or *'love'*.

Image datasets may contain hundreds, thousands or millions of images that are usually annotated with keywords provided by humans. However, images can be automatically annotated (resp., retrieved) using their content, where this content could be a color distribution, texture or forms within the image. Thus, many commercial and public retrieval systems have been built and are already in use to fulfil such tasks. In this chapter, we introduce and discuss various methods used by content-based image retrieval systems.

IV.1 Query Formulation

For a user, there are mainly tow methods to find the images that he is looking for(Kherfi 2008). In the first method, he can browse a catalogue provided by the CBIR interface to find potentially good images. However, such a method might be tedious and time waster. In the second method, User can simply use CBIR interface to formulate a query that describes the desired image. A query can be either a text or an example image. However, using a query is a delicate problem and more difficult than it seems, which arises the following two questions:

1. How one can precisely describes his desired image using only a limited number of tools provided by a CBIR?
2. How CBIR interprets and understands exactly one's desire from a simple query?

Over the past few years, considerable advances have been made to facilitate the interactions between the user and CBIR during the query formulation.

IV.1.1 Query by Text

In this type of queries, users express their needs using words. For example, user that looks for some kind of landscape provides the CBIR with a sentence such as 'mountain with icecap peak surrounded by trees while sun rises'. One challenge with this type of querying is how to analyze the words in order to extract the user's most significant ones. In addition, understanding the exact meaning of the sentence is an extremely hard task, since that the sentence is not a simple set of unlinked words. For example, the word *'Jaguar'* is used for different means and it can be understand, therefore, from the context of the sentence. If one formulates *'find jaguar on tree'* the *CBIR* must be able to understand that the user wants image of jaguar the animal. Whereas, if the query is *'find jaguar on road'* the results must be images of jaguar the car.

Text-based queries for image retrieval offer some pros:

1. Words are natural way used in real-life allowing users to express their selves.
2. Text-search techniques were a subject of research over decades and can be exploited directly in image retrieval.

3. Some concepts, such as *'love'* are extremely hard to be expressed using image example, instead, using text to find similar annotated image is more useful.

Despite the pros text-based querying offers, it suffers from numerous drawbacks:

1. Because people often do not take the time to annotate their photos, most collections does not contain any text along with the images. For this reason, text-query techniques become useless.
2. Different users may annotate the same image with different concepts, which makes it subjective. Actually, annotations expresses the annotator more than the image itself (Jain 1995).
3. A collection that is annotated with one language can be only queried with the same language. An alternative solution might be in translating the query before submitting it.
4. Some text-based image retrievals use the text surrounding the image as an annotation (e.g., images within web pages). However, it is hard to determine which words in the text are relevant and which are not.
5. Text-based querying lacks refinement and may yield ambiguous results. Using the query *'Bolt'*, Figure 22 illustrates the obtained result from Google image. We can see that the retrieval confused *'bolt'* as a dog of the movie, '*Usain Bolt'* the Olympic runner, or *'bolt'* as a metallic tool.

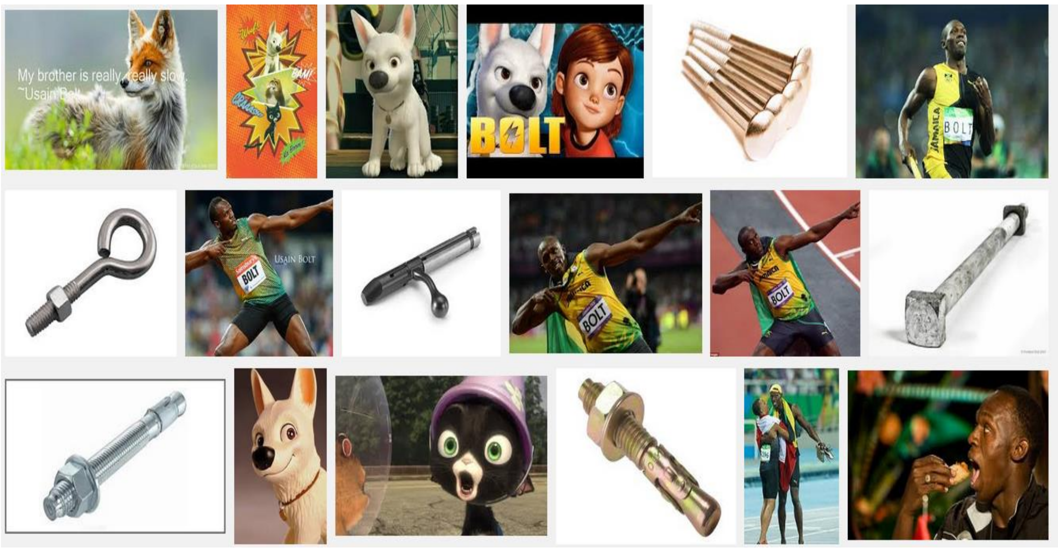


Figure 22. Results obtained from Google image using the word *'bolt'*.

6. It is impossible for text to describe the full details, which demonstrates the limitation of the capacity of text. For example, if one want to find images that are highly similar to the one in Figure 23, it extremely difficult or even impossible to describe such a scene, with high details, using only words.

Figure 23. Illustration of the limitation of text to describe the content of an image.

IV.1.2 Query by Image

As we have seen before, text-based image retrieval has a set of limitations which drive researchers to think whether it is better to let images speak for themselves. Hence, they suggested allowing users to formulate their queries using images instead of words, and then the system has to find other similar images. Indeed, dealing with queries that come in form of image needs different means than the ones used with textual queries. This new method is called content-based image retrieval (CBIR). Next, we will summarize different ways that can be used to formulate a CBIR query. However, one should note that there exist certain common steps for most of the methods, which are:

1. Before putting the CBIR in use, visual descriptors must be extracted from all images of the collection.
2. Generally, CBIR must extract the same visual descriptors from the query.
3. CBIR searches for similar images by comparing the descriptor extracted from the query with those extracted from the collection images.

IV.1.2.1 Using Provided Feature Values

This querying technique asks the user to provide a numerical value for each feature, as in Web-Wise(Wei, Li et al. 1998) image retrieval. For example, if the dataset images are described using color moments or Fourier descriptors, then the user is required to provide a numerical value for these features that corresponds to the desired image. However, such a technique is very difficult to be used by an image-processing specialists rather than an ordinary user. First, an ordinary user may not understand feature meaning, such as color moments or Fourier descriptors. Second, it is very hard, even for a specialist, to turn one's needs (i.e., the image that he is looking for) into a set of numerical values.

IV.1.2.2 Using Example Image

Querying by image example is definitely the most successful and used technique. The principle of this technique is very simple: after submitting an example image by user, the CBIR retrieves and shows the resemble ones. Querying by example image can be summarized as follows:

1. The CBIR starts by proposing to the user a catalogue of images to select from. Alternatively, the user can simply submit, to the CBIR, an example image that he want to find the resemble ones.
2. The CBIR browses through the dataset looking for images that resemble to the query, then, it returns and shows the result to the user in form of sorted list from the most to the least relevant image.

The catalogue of images that CBIR proposes at the beginning might be chosen randomly or intelligently. Thus, choosing these images wisely makes the retrieval more productive and quicker.

IV.1.2.3 Using Sketches and Icons

Some retrievals, such as(Lew 2000), involves granting users to draw a sketch that approximately shows what he is looking for. Creating as well as coloring sketches can be done using a mouse or an electronic pen.

Alternatively, user can use a list of icons (e.g., sun, mountain, tree, car, etc.), pre-proposed by CBIR, in order to design the query. An example of icon-based query CBIR can be found in (Lew 2000).

After discussing the different types of formulating an image query, we find out that CBIR techniques have numerous advantages, including:

1. Unlike text-based image retrieval (TBIR) which requires the dataset to be annotated, CBIR techniques can be exploited even if the dataset does not contain any text.
2. They perform well with images that contain very complex scenes, which can not be described using words only.
3. CBIR results are more objective than TBIR.
4. It allows a higher level of refinement than text. For example, looking for images that visually resemble the one in Figure 23 is quite plausible using example images.

IV.2 Features extraction

"The fact of designing and extracting visual features which accurately represents the content of images is perhaps the pillar of content-based search"(Kherfi 2008). Over decades, many features have been proposed in literature. We have discussed different types of features in Chapter II. However, one should note that features extraction is not a completely resolved problem.

IV.3 Comparing Feature

After extracting the features of the query image, CBIR uses a similarity measures to find similar features and thus similar images in the dataset. Many types of similarity measures have been proposed in literature including:

IV.3.1 Minkowski Distances

They are a set of distance functions that are defined, for two given feature vectors $V$and V', as:

$$d_p(V,V') = \left(\sum_{n=1}^{n}(V_n - V_n')\right)^{\frac{1}{p}} \qquad (43)$$

The well known d_2 (i.e., Euclidean distance) is widely used in computing similarity scores between images. It is a very simple distance that has a low computation cost (i.e., $\mathcal{O}(N)$). Other frequently used Minkowski distances are d_1 (i.e., as Manhattan distance), and d_∞ (i.e., maximum distance).

IV.3.2 Histogram Intersection

It has been proposed by Swain et al.(Swain and Ballard 1991) in order to compare the distance measure between histograms. It consists in calculating the common part of two histograms. It is given by:

$$d_\cap(V,V') = \sum_{n=1}^{N}\min(V_n - V_n') \qquad (44)$$

IV.3.3 Relative Bin Deviation

It calculates the bin-wise deviation between two vectors, Eq. (45) shows relative deviation calculation formula:

$$d_{rbd}(V,V') = \sum_{n=1}^{N}\frac{\sqrt{(V_n - V_n')^2}}{\frac{1}{2}(\sqrt{{V_n}^2} + \sqrt{{V_n'}^2})} \qquad (45)$$

IV.3.4 $\mathcal{X}^2 -$ Distance

$\mathcal{X}^2 -$ distance consists in determining whether two given probabilistic distributions differs or no. it is given by:

$$d_{x^2}(V,V') = \sum_{n=1}^{N} \frac{V_n - V'_n}{V_n + V'_n}.x \tag{46}$$

IV.3.5 Kullback-Leibler Divergence

It is also called discrimination information (Kullback 1987). It has been proposed in information theory to measure the difference between two probability distributions P and Q. It calculates the efficiency of coding one probability using the other. For two given vectors V and V', Kullback-Leibler divergence can be calculated using formula in Eq. (47)

$$d_{KL}(V,V') = \sum_{n=1}^{N} V_n log \frac{V_n}{V'_n} \tag{47}$$

Kullback-Leibler divergence is neither symmetric nor numerically stable.

IV.3.6 Jensen Shannon Divergence

Known as Jeffrey divergence. It is based on the Kullback–Leibler divergence with some notable improvements including that it is symmetric and has a finite value. It is given by Eq. (48).

$$d_{JSD}(V,V') = \sum_{n=1}^{N} V_n log \frac{2V_n}{V_n + V'_n} + V'_n log \frac{2V'_n}{V_n + V'_n} \tag{48}$$

IV.3.7 Bhattacharyya Distance

Also called fidelity. It measures the overlapping extent between two statistical (continuous or discrete) samples or populations. It is defined by the following formula:

$$d_{BCA}(V,V') = \sum_{n=1}^{N} \sqrt{V_n \times V'_n} \qquad (49)$$

Some other extensions of this distance have been proposed in (Nölle 2003), which are :

$$d_{\overline{BCA}}(V,V') = 1 - d_{BCA}(V,V') \qquad (50)$$

$$d_{\sqrt{1-BCA}}(V,V') = \sqrt{1 - d_{BCA}(V,V')} \qquad (51)$$

$$d_{(2-BCA)}(V,V') = Log(2 - d_{BCA}(V,V')) \qquad (52)$$

$$d_{arccos(BCA)}(V,V') = \frac{2}{\pi} arccos(d_{BCA}(V,V')) \qquad (53)$$

$$d_{\sin(BCA)}(V,V') = \sqrt{1 - (d_{BCA}(V,V'))^2} \qquad (54)$$

IV.3.8 Quadratic Forms Distance

Quadratic Forms distance is capable of calculating the similarities between cross-bins. It incorporates a matrix A that denotes the similarity between all possible bin pairs. For given two feature vectors $V$and V', then the Quadratic form distance can be calculated by:

$$d_{qdr}(V,V') = \sqrt{(V - V')^T \cdot A \cdot (V - V')} \qquad (55)$$

where $A_{m,n}$ denotes the similarity value between the two bins *m* and *n*, and it can be calculated by:

$$A_{m,n} = 1 - \frac{d_2(V_m, V_n)}{d_{max}} \qquad (56)$$

where $d_2(V_m, V_n)$ the Euclidean distance between the two vector cells V_m and V_n, and d_{max} is the maximum value of the matrix $(1 - A)$.

IV.3.9 Earth Movers Distance

The Earth Movers Distance (EMD) is a special case of the transportation problem. It consists in calculating the minimum amount of effort that is needed to perform a switch one distribution into the other by moving portions of distributions between the bins. It can be calculated by the following formula.

$$d_{EMD}(V, V') = \frac{\sum_{i,j}(d_{i,j} \times g_{i,j})}{\sum_{i,j} g_{i,j}} \qquad (57)$$

where $d_{i,j}$ is the dissimilarity value between the bins *i* and *j* (i.e., $d_{i,j} = \left|V_i - V'_j\right|$), and $g_{i,j} \geq 0$ is an optimal flow between V and V' such that the total cost is minimum. This flow must be subjected to the following constraints:

a. $\sum_j g_{i,j} \leq V_i$
b. $\sum_i g_{i,j} \leq V'_j$
c. $\sum_{i,j} g_{i,j} = \min(V_i, V'_j)$

The major advantage of EMD is that each image in the collection can represented using a customized binning type.

IV.4 Applications

Billions of images are being generated every increasing rate by many sources such as "defense, civilian satellites, military reconnaissance, surveillance flights, fingerprinting, mug-shot-capturing devices, scientific experiments, biomedical imaging and home entertainment systems". (Gudivada and Raghavan 1995). Therefore, a content-based image retrieval (CBIR) system is required to efficiently use information from these image collections. Such a system helps users to retrieve their desired images based on the content. Application areas in which CBIR is a principal activity are numerous and diverse(Kherfi, Ziou et al. 2004).

IV.4.1 Intellectual Property

One application of CBIR is to grant image copyrights protection and prevent using images without the permission of the original owner. For example, Watermarking is a copyright protection technique that involves including the identity of the creator or distributor to the image. However, Watermarking is vulnerable to geometric distortions, image processing, and subterfuge attacks (Chang, Wang et al. 1998). In addition, the diversity in watermark schemes implies increasing of detection time exponentially. CBIR might be an excellent alternative solution to such a problem. Chang et al. (Chang, Wang et al. 1998) has proposed an example of a CBIR system designed to detect replicated images on the Web.

IV.4.2 Filtering of Inappropriate Mature Content

Pornographic images and videos have invaded a large number of websites. Because of internet is available anywhere and accessible by children, pornographic images became a problem that many parents are concerned about. In addition, such images are undesired even for adults in some cultures. Therefore, there is an urgent need for a tool for filtering the inappropriate mature from the internet. Traditional solutions that try to prevent children from accessing undesirable websites are filter software such as NetNanny, Cyber Patrol, and CyberSitter (Wang, Li et al. 1998). These software use *ip* addresses or the text within documents to make their decision. Therefore, they are ineffective in most cases because pornography websites often contain more images and less text. Computer vision techniques have been explored to automatically identify pornographic images. For example, authors of (Fleck, Forsyth et al. 1996, Cao, Li et al. 2002) have proposed an algorithm that identifies images containing naked people. They use texture and color for skin filtering. A similar system was developed by (Chan, Harvey et al. 1999) to identify images of naked people using skin tone and limb shape. Integrating such tools in web CBIR helps considerably in identifying and filtering suspicious sites. IBCOW is a website classifier proposed by (Wang, Li et al. 1998). It simply scans then classifies websites into *"objectionable"* or *"benign"* based on their image content.

IV.4.3 Law Enforcement and Crime Prevention

Law enforcement and crime prevention have exploited CBIR for many purposes, such as fingerprint recognition (Marasco and Ross 2015), face recognition (Cevikalp and Triggs 2010), DNA matching, shoe sole impressions (Bijhold and Geradts 2002) and surveillance systems (Eakins and Graham 1999). Internet can be used by criminals as a mean to promote their illicit goods and services such as illegal weapons and drugs. The existing of more visual information and less text in many websites prevents locating them relying exclusively on text retrieval techniques. Thus, integrating content-based with text-based techniques clearly makes the tool vital and able to locate such websites. Other illegal operations include using internet to call for violence, racism, and Nazism.

IV.4.4 Travel and Tourism

People always want to know more about the new places before visiting them. They might want to see the map of the destination country, transportation networks, monuments and tourist attractions. In addition, tourists may also be interested in gaining a basic idea about the country’s way of life such as its markets, characteristic architecture, and traditional dress. Images may be the best means to provide such information. Thus, there is a need for an automatic retrieval to help tourists and travelers benefit.

IV.4.5 Education and Training

Another important application field of image retrieval is education (Wactlar, Kanade et al. 1996, Van Der Zwan, Kukulska-Hulme et al. 1999). Students regularly need images to do research on particular subjects for two main purposes:

1. As a source of information (e.g., a map of Algeria in 1980) or to show some cultural ideas (e.g., custom images of Algerian Tuareg).
2. Teachers can also use these images to prepare the courses, and integrate them with other academic materials to better illustrate and explain the ideas.

Such images are freely available on the World Wide Web. However, locating them manually is a very tedious and time-consumer task. Having a Web CBIR makes this task more reliable and easier.

IV.4.6 Home Entertainment

Internet provides means of entertainment for many people, such as joke images, caricatures, comic strips, movie shots, and music clips. However, finding these images or the visual media that the people are looking for is often a trouble. Therefore, creating image and video retrieval tools for such purposes seems to be indispensable in helping users locate the sought images. Furthermore, cataloguing the visual content in certain index structure makes navigation to such materials easier and enjoyable.

IV.4.7 Fashion, Architecture, and Engineering Design

International designers of graphics, fashions and industrials share their images of previous designs as a source of inspiration(Yang, Garrett Jr et al. 1994, Bird, Elliott et al. 1996). When developing new projects, architects and engineers may need to visualize plans, machine images and other related material. With the appearance of internet, these professionals become no longer limited to the local collections in their inspiration quest. The internet allows them to access other images and videos that may seem similar or different styles providing richer stimulation for their imagination. Thus, CBIR systems and browsing catalogs are useful in all these fields.

IV.4.8 Historical and Art Research

Visual data can also be used by historians, archaeologists, and sociologists as a source of information or to support their research. Sometimes, accessing to original works of art is restricted (e.g., due to geographic gap, ownership constraints, or work physical-condition), researchers can use alternatives, which are found on the Web in the form of photographs or pictures of objects, to avoid this problem. A Web CBIR can save them time in looking for such materials. Examples of CBIR systems attempt to apply image retrieval techniques to art and historical research include (Jain, Goel et al. 1997, Barnard and Forsyth 2001, Chen and Wang 2002).

IV.5 CBIR Systems

There are many CBIR prototypes proposed in recent years. In this section, we list and discuss the principles of some of these CBIR systems. It should be noted that most of the reviewed CBIR systems are prototypes. Some of them however do offer online demos.

IV.5.1 Query by Image Content System

'IBM Almaden Research Center, San Jose, CA' has developed Query by Image Content (QBIC)(Niblack, Barber et al. 1993) as a project studying methods to extend and complement TBIR by retrieving images and videos based on their content. As image features, QBIC uses:

1. *Color:* the computed color features are, the 3D average color vector of an object/full-image, and a 256-dimensional RGB color histogram in the color spaces: RGB, YIQ (luma, in-phase, quadrature), Lab (luminance, a-axes, b-axes) and Munsell.
2. *Texture:* QBIC uses modified versions of the coarseness, contrast, and directionality of Tamura features.
3. *Shape:* The shape features consist of shape area, circularity, eccentricity, major axis orientation and a set of algebraic moment invariants extracted from object contours.

QBIC allows queries based on example images, user-designed sketches and selected color and texture patterns. In the latter case, QBIC offers samplers to the user to choose its colors and textures from. The proportion of the desired color in the image is adjusted while moving sliders.

IV.5.2 Netra

'Department of Electrical and Computer Engineering, University of California' has developed Netra(Ma and Manjunath 1999) image retrieval prototype that uses color, texture,

shape and spatial location features of segmented image regions of homogeneous colors to find and retrieve similar regions from the dataset. As features, Netra uses:

1. *Color:* After quantizing and representing the RGB color space using 256 bins codebook, Netra extracts, for each image region c, a color vector $f_c = (c_0, p_0, c_1, p_1, \dots c_n, p_n)$ where c_i is the index of bin i in the codebook, p_i is the proportion of that color in the region c, $p_0 + p_1 + \cdots p_n = 1$, and n is the number of used colors to represent the corresponding region.
2. *Texture:* texture has been represented by a feature vector f_t that contains normalized mean and standard deviation of a series of Gabor wavelet transformations of the image: $f_t = (\mu_{0,0}, \dots \mu_{s,k}, \sigma_{0,0}, \dots, \sigma_{s,k})$ where s indicates the scales number and k is the directions number.
3. *Shape:* three feature vectors have been used to represent the shape information of a region namely f_K, f_R andf_Z which represent, respectively, the curvature at each point on the contour, the distance between each contour's point and centroid of the region, and each point as a complex number with real part equal to the *x*-coordinate, and the imaginary part equal to the *y*-coordinate.

Netra allows querying using a catalogue of images it offers. Alternatively, User can simply designed sketches using color and texture patterns. The proportion of the desired color in the image is adjusted while moving a slider. All images in the dataset have been segmented into homogeneous regions. Thus, User can simply click on one of the regions and select one of the four features color, spatial location, texture or shape.

IV.5.3 VisualSEEk

'Image and Advanced Television Lab, Columbia University' has developed a highly functional prototype system for visual features searching in image datasets. VisualSEEk(Smith and Chang 1997) finds images that contain the most similar-regions arrangements. It uses absolute and relative spatial color information, region centroid along with the width and height of the minimum bounding region rectangle.

To create a query, the user sketches some regions, and then locates and set dimensions for them on a given grid, and finally selects a color for each region. In addition, the user can indicate the boundaries of locations, sizes and spatial relationships between the regions. After returning a preliminary result of images, the user is then able to search by example using the returned images.

IV.5.4 MARS

'Department of Computer Science, University of Illinois' started a project to design and develop an integrated multimedia information retrieval and dataset management infrastructure, entitled Multimedia Analysis and Retrieval System (MARS) (Ortega, Rui et al. 1997). Specifically, MARS project can be categorized into multimedia content representation, multimedia information retrieval, multimedia feature indexing and multimedia database management. It supports querying by combining low-level features (i.e., color, texture, shape) and textual descriptions.

1. *Color:* it is represented by a 2D histogram over the HS channels of the HSV color space.
2. *Texture: it* is represented using two histograms, the first one measures the coarseness and the other one measures the directionality of the image, In addition to one scalar that defines the image contrast.
3. *Shape:* The boundary shape of the extracted object is represented using means of Fourier descriptors.

In MARS, User can formulate complex queries using boolean operators. The desired features can either be specified using example (i.e., pointing an image that has such a property) or direct (i.e., by choosing colors or textures from available palette or a set of patterns).

IV.5.5 PicToSeek

'Department of Computer Science, University of Amsterdam' has proposed then integrated an image retrieval scheme into their PicToSeek system(Gevers and Smeulders 2000). The developers define color and shape invariants to be used as features, in the proposed system, independent of camera viewpoint, object's geometry and illumination conditions. They analyzed the effect of various imaging conditions on the different color features, under the assumption of a dichromatic reflection model. As features, PicToSeek uses:

1. *Color:* represented by three histograms H_a, H_b and H_c, where H_a and H_b are computed based on a proposed color model (i.e., coded from the RGB color space), and H_c is histogram of color edges.
2. *Shape:* represented by two histograms H_D and H_E. The first one, is constructed on a discretized angle (i.e., angle between two color edges) while the latter on the cross ratio (i.e., cross-ratio between four color-edges originating in the same point).

A query image can be selected from a catalogue proposed by the system or simply by submitting an example image. User is also allowed to directly provide feature values of the desired image.

IV.6 Conclusion

In this chapter, we introduce, explain and discuss in details the content-based image retrieval. We firstly see how CBIR operates, how query types (i.e., by text and by image) differs, and what query by image offers as pros. Then, we list a set of distance measures that can be used to calculate the similarity between features. Then, we explain how there is an ergent need for CBIRs in different application such as intellectual property, law enforcement, travel and tourism, education, etc. Finally, we list some CBIR prototypes that are online, operational, and public.

Chapter V.MODIFIED INTEGRATIVE COLOR INTENSITY CO-OCCURRENCE MATRIX

As explained in Section III.3, Vadivel et al.(Vadivel, Sural et al. 2005, Vadivel, Sural et al. 2007) have succeeded to calculate the extent to which a given pixel is a color or gray level. However, their method uses a hard mapping scheme to assign a given color (or gray level) to the right bin, which represents a serious weakness. To illustrate this, let us take the example of Figure 24. The color quantization and mapping process they used considers that all the hue values *p1, p2* and *p3* are pure magentas, which is not precise. Indeed, these quantities represent different colors: *p1* is a reddish magenta, *p2* is a pure magenta and p3 is a bluish magenta. Even though a quantization process can attributes them to the same bin, they should not be considered to be the same, otherwise, a considerable amount of information about colors could be falsely interpreted.

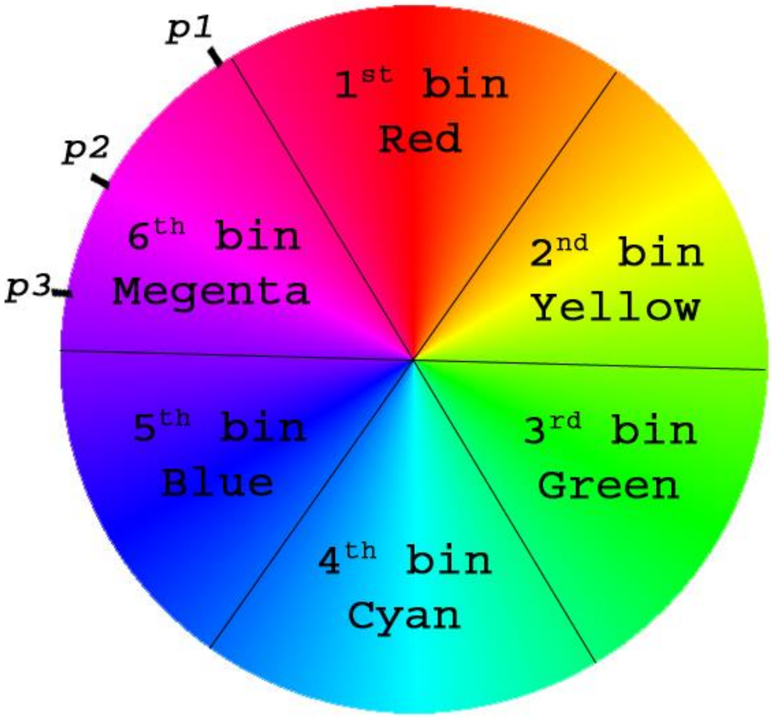


Figure 24. The hard mapping used by Vadivel et al. considers that the three colors *p1*, *p2* and *p3* as pure magentas. However, *p1* is a 55% magenta and 45% red; p3 is 60% magenta and 40% blue.

Now, we will explain the steps of the method we propose, and show how it overcomes the raised issue. Given a color image *I*, we first transfer it into HSV color space, we then perform color mapping and gray-level mapping of each pixel, and finally, we extract our new representation MICICM (Khaldi and Kherfi 2016). These steps are detailed in the following three sub-sections.

V.1 HSV Color Space Analysis

Hue, Saturation, Value (HSV) color model describes colors (tint) in terms of their shade (amount of gray) and brightness (luminance). Unlike RGB color model (Tan and Kittler 1994, Wandell 1996), HSV channels are totally di-correlated which make it analogues to human vision. In other words, all tints, tones and shades of the some specific color have the same *hue*.

V.1.1 HSV Model

Usually, HSV color model is described using a *hexcone* as it is illustrated in Figure 25. It has the following properties:

a. It describes *hue* by a number $h\epsilon[0\ 1]$ (resp., $[0\ 2\pi]$) that specifies the position of the corresponding pure color on the color wheel. For example, $h = 0$ refers to red, $h = {}^{2}/_{3}$ is blue and $h = {}^{5}/_{6}$ is magenta, and so forth.

b. It describes *saturation* by a number $s\epsilon[0\ 1]$ that specifies the position of the corresponding proportion of white in the color on wheel's radius. For example, pure red has a saturation of 1, tints of red have saturations less than 1 and white has a saturation of 0.

c. It describes *value* by a number $v\epsilon[0\ 1]$ that specifies the position of the corresponding proportion of darkness, in the color, on hexcone's vertical exes. It is also called lightness. For example, a value of 0 means that the color is black regardless its *hue* and *saturation*, with increasing in lightness color moves away from black.

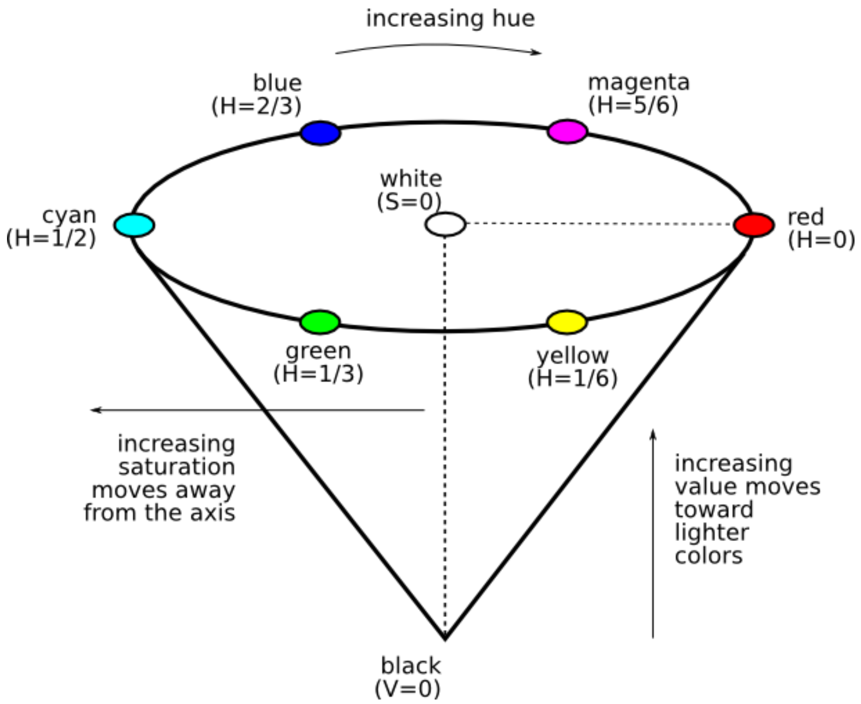


Figure 25. single-hexcone model of HSV color space.

V.1.2 Converting RGB to HSV color space

Because the channels of RGB color-space are highly correlated with each other, describing images in terms of those channels makes their discrimination difficult. Hence, applying algorithms such as k-means(MacQueen 1967, Lloyd 1982), fuzzy clustering(Dunn 1973, Bezdek 2013) or canny edge detection(Canny 1986), where each channel passed on the same algorithm, is useless because of the correlation between those channels. Therefore, transforming images from RGB into HSV color space helps to eliminate the correlation between these channels and permit to use the latter algorithms. An RGB color can be transformed into HSV color space using the following set of formulas (Eq. (58)-(60)).

$$H = \begin{cases} 0° & if\ \Delta = 0 \\ 60° \times \left(\frac{G' - B'}{\Delta}\ mod\ 6\right) & if\ C_{max} = R' \\ 60° \times \left(\frac{B' - R'}{\Delta} + 2\right) & if\ C_{max} = G' \\ 60° \times \left(\frac{R' - G'}{\Delta} + 4\right) & if\ C_{max} = B' \end{cases} \tag{58}$$

$$S = \begin{cases} 0 & if\ C_{max} = 0 \\ \frac{\Delta}{C_{max}} & if\ C_{max} \neq 0 \end{cases} \tag{59}$$

$$V = C_{max} \tag{60}$$

where $R' = R/255$, $G' = G/255$, $B' = B/255$, $C_{max} = \max(R', G', B')$ and $\Delta = C_{max} - \min(R', G', B')$.

After having all pixels of the image converted into HSV color space, they have to be mapped to their appropriate bins before start extracting the MICICM. In the next sub-section, we introduce our proposed fuzzy mapping scheme.

V.2 Color mapping and gray-level mapping

As we discussed above, the main drawback of the hard mapping is that it assigns each hue value to a unique bin and each gray-level value to a unique bin too. In order to avoid this drawback, we have developed a new mapping scheme where each color is mapped to two bins with different weights and each gray level to two bins with their weighs too. Indeed, most colors are not pure but rather result from the mixing of their true colors with the ones of their neighboring colors (either the left or the right one). Let us take the same example as above (i.e., Figure 24). The hard mapping has considered *p1* as a pure magenta. However, a careful inspection shows that it rather comes from mixing magenta (its true hue bin) with red (the right-neighbor hue bin). Consequently, our mapping will assign it to both bins: red and magenta. In addition, it will calculate the contribution of magenta in producing *p1*, as well as the contribution of red. Similarly, we can say that *p3* comes from mixing magenta (its true hue bin) with blue (the left-neighbor hue bin). Our mapping will therefore assign it to both bins: magenta and blue, and calculate the contribution of each to it. Finally, we notice that the right neighbor and the left neighbor hue bins should not participate together in producing the same color. If the color is situated in the left half of its true hue bin, then the left neighbor bin will participate in producing it. Otherwise, it is the right neighbor bin which will participate.

In a similar way, we can say that each gray level may be considered as a combination of two bins which have participated in producing it (its true gray-level bin and one of the neighbors). Therefore, it will be assigned to both bins and the contribution of each bin will be calculated.

In short, given a pixel with a hue h and a gray-level g, our mapping will calculate:

1. $Wc(h)$ and $Wc_N(h)$, which, respectively, are the contribution of the true hue bin of h and the contribution of its neighbor hue bin.
2. $Wg(g)$ and $Wg_N(g)$ which, respectively, are the contribution of the true gray-level bin of g and the contribution of its neighbor gray-level bin.

We want the weights $Wc(h)$ and $Wc_N(h)$ to satisfy the following constraints:

a. $Wc(h)\in [\frac{1}{2},1]\ and\ Wc_N(h)\in [0,\frac{1}{2}]$.

b. $Wc_N(h) + Wc(h) = 1.$

c. If h is located in the left half of its true hue bin, then it is its left neighbor bin which contributes, otherwise, if h is located in the right half, then it is its right neighbor which contributes.

d. The contribution is significant only if h is located near the boundary, and then it decreases quickly to stabilise around zero as soon as h moves away from the boundary to the center of its true hue bin.

Several functions may satisfy the constraints (a-d) stated above. After a detailed analysis, we found that the following functions (Eq. (61) and Eq. (62)) are quite satisfactory:

$$Wc_N(h) = \begin{cases} \frac{1}{2} e^{-\left|t1\frac{h- H_0}{HBinSize}\right|^{t2}} & ; if\ the\ contributor\ neighbor\ is\ the\ left\ bin \\ \frac{1}{2} e^{-\left|t1\frac{h- H_1}{HBinSize}\right|^{t2}} & ; if \text{ the } contributor\ neighbor\ is\ the\ right\ bin \end{cases} \tag{61}$$

$$Wc(h) = 1 - Wc_N(h) \tag{62}$$

where *t1* and *t2* are adjustment parameters. *HBinSize* is the size of a bin after sampling the hue channel. H_0 is the smallest hue value of the bin in which *h* falls, and *H1* is the greatest hue value of the same bin (*H1* = *H0* + *HBinSize*).

In a similar way, we want the weights related to the gray-level perception to satisfy similar constraints. We therefore opted for the following functions (Eq. (63) and Eq. (64)):

$$Wg_N(g) = \begin{cases} \frac{1}{2} e^{-\left|k1\frac{g- G_0}{GBinSize}\right|^{k2}} & ; if\ the\ contributor\ neighbor\ is\ the\ left\ bin \\ \frac{1}{2} e^{-\left|k1\frac{g- G_1}{GBinSize}\right|^{k2}} & ; if \text{ the } contributor\ neighbor\ is\ the\ right\ bin \end{cases} \tag{63}$$

$$Wg(g) = 1 - Wg_N(g) \tag{64}$$

where *k1* and *k2* are adjustment parameters. *GBinSize* is the size of a bin after sampling the gray-level channel. G_0 is the smallest gray-level value of the bin in which *g* falls and *G1* is the greatest gray-level value of the same bin (*G1* = *G0* + *GBinsize*).

Unlike FCCM(Ledoux, Losson et al. 2015), our function offers more flexibility and it better controls the range of contribution of neighbours to the color (resp., gray level). *t1* controls the max range that a neighbour color could reach in its contribution, whereas, t2 controls the softness the function within the defined range as Figure 26 shows. However, one should note that for a big values of t1 and t2 the proposed function becomes a hard mapping, whereas, for too small values the contribution of the neighbour may accurses even if the color is pure. Therefore, a suitable values for these two parameters should be empirically tuned as we did in VI.2.

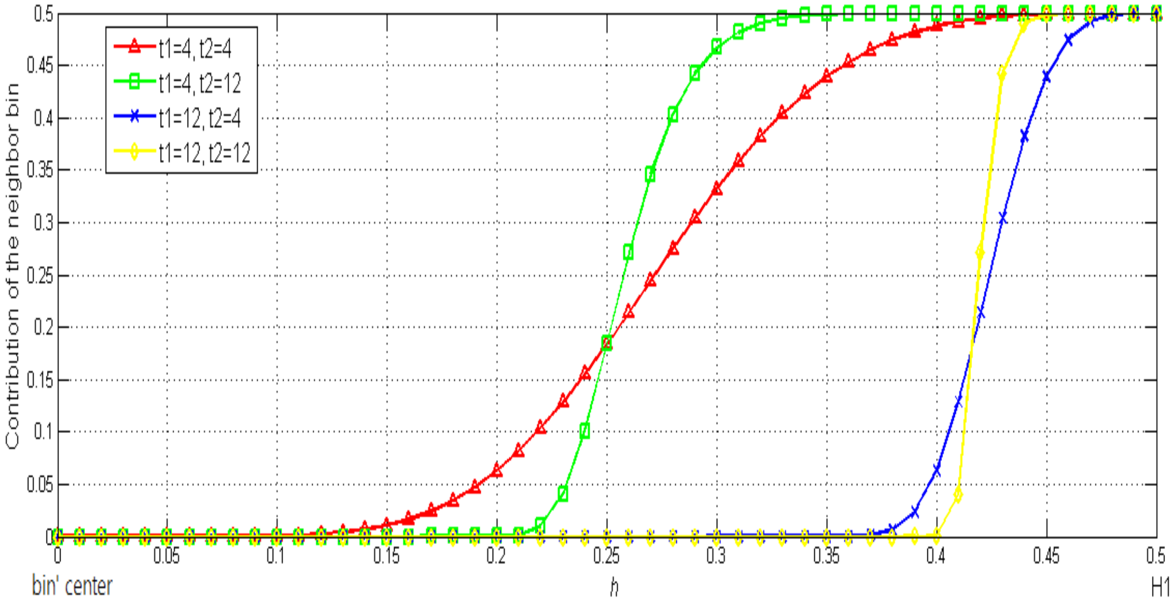


Figure 26. Behaviors of our mapping function $Wc_N(h)$ (resp. $Wg_N(g)$) for different values of parameters t1 and t2.

V.3 Extracting the Co-occurrence Matrix

Once colors and gray levels are mapped, we extract our MICICM. It is constituted of four sub-matrices as illustrated in Figure 27, where:

- $MICICM_{cc}$ holds information about the co-occurrence of color-color pixel pairs.
- $MICICM_{cg}$ holds information about the co-occurrence of color-gray pairs.
- $MICICM_{gc}$ holds information about the co-occurrence of gray-color pairs.
- $MICICM_{gg}$ holds information about the co-occurrence of gray-gray pairs.

For example, MICICM(*c1*,*c2*) is the occurrence frequency of the hue pairs (*h1*, *h2)* such that the mapping result of *h1* and *h2* are, respectively, the bins *c1* and *c2*.

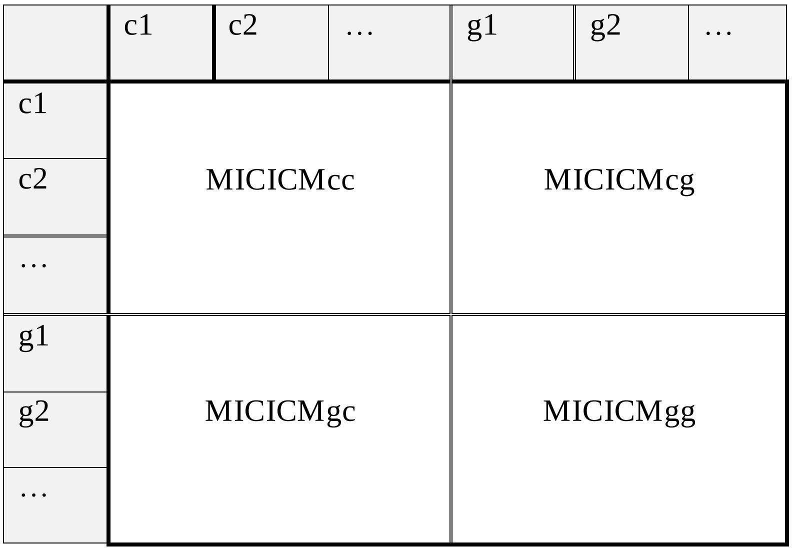


Figure 27. General view of MICICM

Although our MICICM contains four sub-matrices like the ICICM of(Vadivel, Sural et al. 2007) does, there is a significant difference between them. In(Vadivel, Sural et al. 2007), each pixel is mapped to a unique color bin and to a unique gray-level bin. Therefore, a given pixel *p1* and its neighbor *p2* will contribute to only one cell in *ICICMcc*, one cell in *ICICMcg*, one cell in *ICICMgc* and one cell in *ICICMgg*. In our case, however, each pixel is mapped to two contributing color bins and two contributing gray-level bins. Since then, *p1* and its neighbor *p2* will contribute to four cells $(i.e., 2^2 = 4)$ in *MICICMcc*, four cells in *MICICMcg*, four cells in *MICICMgc* and four cells in *MICICMgg*.

In order to make this latter point clearer, let us take the example of how they contribute to four cells in the first sub-matrix *MICICMcc*. First, Assume that *p1* is a given pixel located at the point (x, y) in the image that has a hue value h_1 mapped to the two hue bins *Qc1* and *QcN*. Second, *p2* is another pixel which co-courses at a distance *d* from p1 *(*i.e., at the point $(x + \Delta x, y + \Delta y))$ that has a hue value h_2 mapped to the two hue bins *Qc2* and *QcN2*, as it is illustrated in Figure 28. Then, the pair (*p1*, *p2*) will contribute to the four following cells: *MICICMcc*(*Qc1*, *Qc2*), *MICICMcc*(*Qc1*, *QcN2*), *MICICMcc*(*QcN1*, *Qc2*) and *MICICMcc*(*QcN1*, *QcN2*).

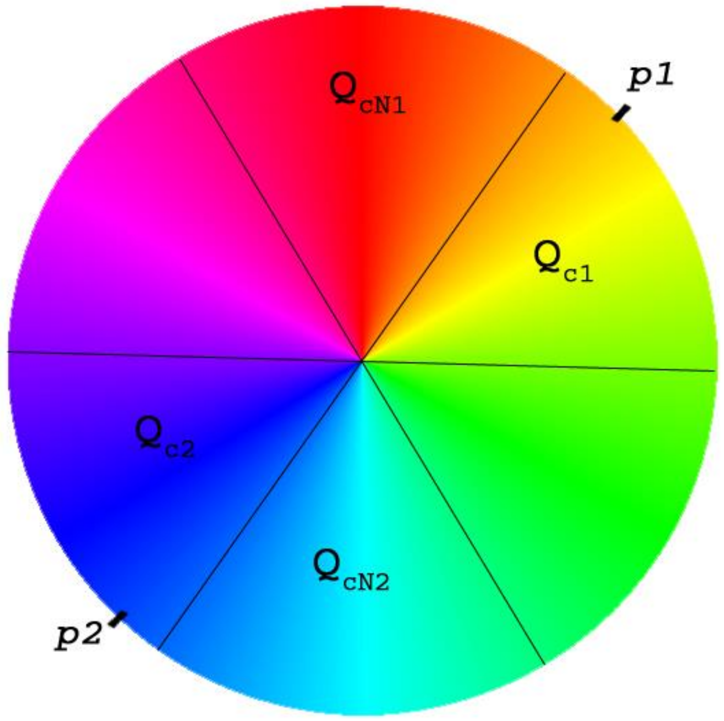


Figure 28. Example of our proposed pixel mapping scheme. *p1* is mapped to both the true hue bin *Qc1* and the left-neighbor hue bin *QcN1* (i.e., yellow and red); *p2* is mapped to both the true hue bin *Qc2* and the left-neighbor hue *QcN2* (i.e., blue and Cyan).

The contribution of *p1* and *p2* to the first cell *MICICMcc*(*Qc1*, *Qc2*) should be reciprocally proportional to the weights *Wc1* and *Wc2* which, respectively, are the contributions of the true hue bins (i.e., *Qc1* and *Qc2*) to produce *p1* and *p2*. It also should be reciprocally proportional to the weights *Wcol1* and *Wcol2* which, respectively, are the color extent of *p1* and the color extent of *p2*. We, therefore, take the product of these four quantities (i.e., *Wc1, Wc2, Wcol1* and *Wcol2*) to update MICICMcc(*Qc1*, *Qc2*) such that $MICICMcc(Qc1, Qc2) += (Wc1 \times Wcol1) \times (Wc2 \times Wcol2)$. Similarly, we obtain the contribution to the three other cells of $MICICMcc$.

In the same way, *MICICMcg*, *MICICMgc* and *MICICMgg* are extracted. For a given image *I*, Algorithm 2 summarizes the needed different steps to, first, map both color and gray-level values, and then, extract the *MICICM*, where:

Wc1 : is the contribution of the true hue bin to produce p1.

Wcol1 : is the color extent of p1.

Wc2 : is the contribution of the true hue bin to produce p2.

Wcol2 : is the color extent of p2.

WcN1 : is the contribution of the neighbor hue bin to produce p1.

WcN2 : is the contribution of the neighbor hue bin to produce p2.

Wg2 : is the contribution of the true gray-level bin to produce p2.

WgN2 : is the contribution of the neighbor gray-level bin to produce p2.

Wgray2: is the gray-level extent of p2.

Wg1 : is the contribution of the true gray-level bin to produce p1.

WgN1 : is the contribution of the neighbor gray-level bin to produce p1.

Wgray1: is the gray level extent of p1.

+= : stands for incrementing variables.

Algorithm 2: Color mapping, gray-level mapping and MICICM calculation

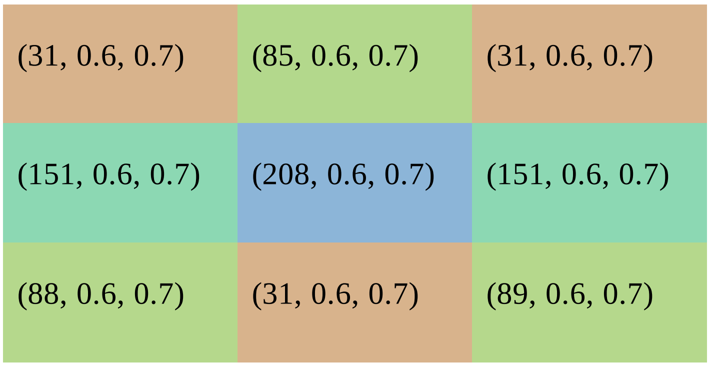
Input: An image *I*

Output: The *MICICM* matrix

```
▷ First step: Color mapping and gray level mapping
foreach pixel $px$ which has a hue value $h$ and gray-level value $g$ in $I$
    Compute its color extent $W_{col}(px)$ and its gray-level extent $W_{gray}(px)$ according to Eq. (38) and Eq. (39).
    Determine the two color bins that contribute to this pixel, and calculate their respective contributions $Wc(h)$ and $Wc_N(h)$ according to Eq. (62) and Eq. (61).
    Determine the two gray-level bins that contribute to this pixel, and calculate their respective contributions $Wg(g)$ and $Wg_N(g)$ according to Eq. (64) and Eq. (63).
end foreach
▷ Second step: calculating MICICM.
**foreach** co-occurrence of pixels (*px1*, *px2*) **in** *I*
    ▷ Calculating $MICICMcc$
    $MICICMcc[Qc1, Qc2] \mathrel{+}= (Wc1 \times Wcol1) \times (Wc2 \times Wcol2)$
    $MICICMcc[Qc1, QcN2] \mathrel{+}= (Wc1 \times Wcol1) \times (WcN2 \times Wcol2)$
    $MICICMcc[QcN1, Qc2] \mathrel{+}= (WcN1 \times Wcol1) \times (Wc2 \times Wcol2)$
    $MICICMcc[QcN1, QcN2] \mathrel{+}= (WcN1 \times Wcol1) \times (WcN2 \times Wcol2)$
    ▷ Calculating $MICICMcg$
    $MICICMcg[Qc1, Qg2] \mathrel{+}= (Wc1 \times Wcol1) \times (Wg2 \times Wgray2)$
    $MICICMcg[Qc1, QgN2] \mathrel{+}= (Wc1 \times Wcol1) \times (WgN2 \times Wgray2)$
    $MICICMcg[QcN1, Qg2] \mathrel{+}= (WcN1 \times Wcol1) \times (Wg2 \times Wgray2)$
    $MICICMcg[QcN1, QgN2] \mathrel{+}= (WcN1 \times Wcol1) \times (WgN2 \times Wgray2)$
    ▷ Calculating $MICICMgc$
    $MICICMgc[Qg1, Qc2] \mathrel{+}= (Wg1 \times Wgray1) \times (Wc2 \times Wcol2)$
    $MICICMgc[Qg1, QcN2] \mathrel{+}= (Wg1 \times Wgray1) \times (WcN2 \times Wcol2)$
    $MICICMgc[QgN1, Qc2] \mathrel{+}= (WgN1 \times Wgray1) \times (Wc2 \times Wcol2)$
    $MICICMgc[QgN1, QcN2] \mathrel{+}= (WgN1 \times Wgray1) \times (WcN2 \times Wcol2)$
    ▷ Calculating $MICICMgg$
    $MICICMgg[Qg1, Qg2] \mathrel{+}= (Wg1 \times Wgray1) \times (Wg2 \times Wgray2)$
    $MICICMgg[Qg1, QgN2] \mathrel{+}= (Wg1 \times Wgray1) \times (WgN2 \times Wgray2)$
    $MICICMgg[QgN1, Qg2] \mathrel{+}= (WgN1 \times Wgray1) \times (Wg2 \times Wgray2)$
    $MICICMgg[QgN1, QgN2] \mathrel{+}= (WgN1 \times Wgray1) \times (WgN2 \times Wgray2)$
end foreach
```

V.4 A Detailed Example

Let us take the following example in order to illustrate the different steps of our algorithm. We want to extract the MICICM from the image *I* represented in Figure 29, where each cell contains respectively the hue, saturation and value of the corresponding pixel. For example *I(1, 1) = (31, 0.6, 0.7)* means that the pixel *I (1, 1)* has a hue of *31*, a saturation of *0.6* and a value of *0.7*. In our example, we gave all the pixels the same saturation (which equals *0.6*) and the same intensity (which equals *0.7*). This has been deliberately done in order to alleviate calculations and allow the reader to focus his attention on the hue which varies from pixel to pixel. Furthermore, we will limit ourselves to the calculation of the first sub-matrix *MICICMcc*. The other sub-matrices are extracted in a similar way.



(31, 0.6, 0.7)	(85, 0.6, 0.7)	(31, 0.6, 0.7)
(151, 0.6, 0.7)	(208, 0.6, 0.7)	(151, 0.6, 0.7)
(88, 0.6, 0.7)	(31, 0.6, 0.7)	(89, 0.6, 0.7)

Figure 29. An example image represented in the HSV space. The three values in each pixel correspond respectively to hue, saturation and value.

Assume that the hue channel has been quantized into six bins namely, *H1*, *H2*, *H3*, *H4*, *H5* and *H6*, as illustrated in Figure 30.

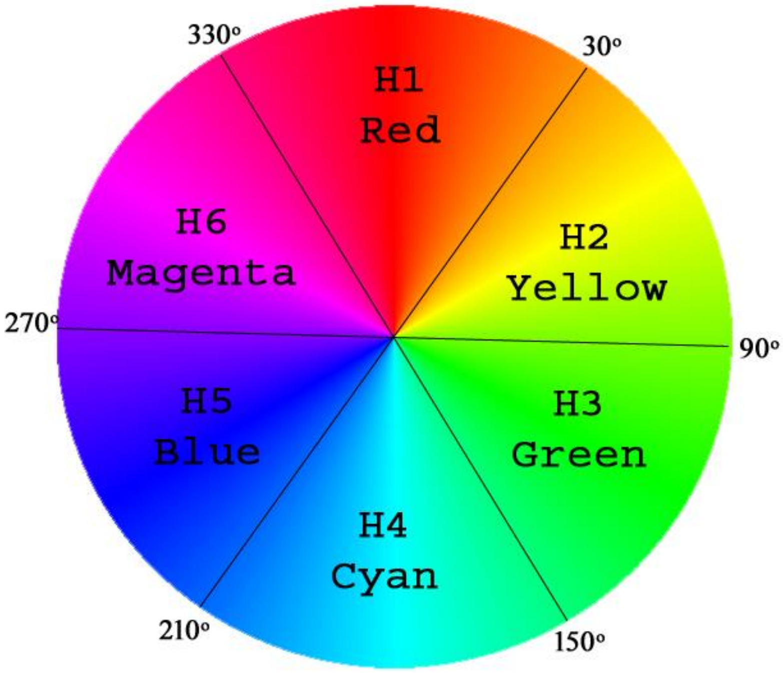


Figure 30. Quantizing hue channel into six bins labelled *H1* to *H6*.

V.4.1 First Step: Calculating Color Extent and Gray-Level Extent

We firstly use Eq. (38) and Eq. (39) to calculate the color extent and the gray-level extent of each pixel. The parameters *t1* and *t2* has been set to *0.1* and *0.85* respectively, which are the optimal values given in (Vadivel, Sural et al. 2007). Since *saturation* and *intensity* are the same for all pixels, these latter will obtain the same color extent *Wcol* and the same gray-level extent *Wgray*:

- $W_{col}(0.6,\ 0.7) = 0.6^{0.1\times\left(\frac{1}{0.7}\right)^{0.85}} = 0.93$
- $W_{gray}(0.6, 0.7) = 1 - \ 0.93 = 0.07$

V.4.2 Second Step: Hue Mapping

Since we are dealing with *MICICMcc* in this example, then we should identify the two hue bins (i.e., the true hue bin and its neighbor hue bin) that contribute to each pixel and calculate their respective contributions. Let us consider the top-left pixel *I(1,1)*. According to our binning, and since its hue equals 31^o, then it falls in the bin *H2* (i.e., *Qc (31) = 2*) which represents its true hue bin. In addition, since it belongs to the left half of this bin, then the neighbor hue bin which contribute to it is the bin *H1* (i.e., *QcN(31) = 1*). Now, according to Eq. (61) and Eq. (62), the contribution of the left neighbor hue bin and that of the true hue bin are respectively:

$$\mathrm{Wc_N}(31) = \frac{1}{2}\mathrm{e}^{-\left(5\frac{31-1}{60}\right)^2} \approx 0.49$$

$$Wc(31) = 1 - Wc_N(31) = 1 - 0.49 = 0.51$$

These results are interpreted as follows: the hue of this pixel is a mixture of the 1st bin (red) and the 2nd bin (yellow) with respective contributions of 0.49 and 0.51.

The other pixels are mapped in the same way, which gives the mapping matrix given in Fig. (32).a. For the sake of comparison, we give in Fig. (32).b the results of the mapping done by the ICICM. We notice that this method map each hue value to a unique bin regardless its position within this bin, which is imprecise. As an example, the pixel *I(1, 1)* has been mapped to the 2nd bin (yellow) although it is very close to the 1st bin (red).

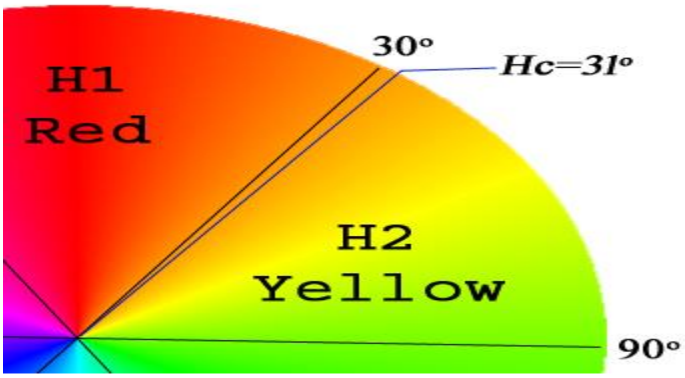


Figure 31. The hue *Hc=31°* is a yellow that is located near the boundaries of red. For MICICM, *Hc* could be considered as both yellow and red. For ICICM however, it is considered as yellow only.

H1 bin: 0.49 *H2* bin: 0.51	*H3* bin: 0.41 *H2* bin: 0.59	*H1* bin: 0.45 *H2* bin: 0.55
H3 bin: 0.41 *H4* bin: 0.59	*H5* bin: 0.46 *H4* bin: 0.54	*H3* bin: 0.48 *H4* bin: 0.52
H3 bin: 0.46 *H2* bin: 0.54	*H1* bin: 0.48 *H2* bin: 0.52	*H3* bin: 0.48 *H2* bin: 0.52

(a)

H2 bin	*H2* bin	*H2* bin
H4 bin	*H4* bin	*H4* bin
H2 bin	*H2* bin	*H2* bin

(b)

Figure 32. Results of hue channel mapping. The left part (a) is hue mapping based on our proposed scheme, where each value is represented by the contributing bins and their corresponding weights. For example the pixel *I(1, 1)* is mapped to the two bins *H1* and *H2* with respective weights 0.49 and 0.51. The right part (b) is a hue mapping based on ICICM, where each element is mapped to one bin only. For example the pixel *I(1, 1)* is mapped to the bin *H2*.

V.4.3 Third Step: Calculating *MICICMcc*

Now that the mapping is done, we extract our MICICMcc according to Algorithm 2. Let us take the example of the pair of pixels $\left(I(1,1),\ I(1,2)\right)$ to see how it contributes to the different cells of this matrix. From Figure 32.(a), the two bins that contribute to *I(1,1)* are *H1* and *H2*, with respective contributions of 0.49 and 0.51, and the two bins that contribute to *I(1,2)* are the *H2* and the *H3* with respective contributions of 0.59 and 0.41. Furthermore, we know that color extent of both pixels equals 0.93. Therefore, this pair will contribute to the four following cells within our matrix: MICICMcc(1,2), MICICMcc(1,3), MICICMcc(2,2) and MICICMcc(2,3). The four cells will be updated according to the lines 10-13 of Algorithm 2 as follows:

$$MICICMcc[1,2] += (0.49 \ \times 0.93) \times (0.59 \times 0.93) = 0.25$$

$$MICICMcc[1,3] += (0.49 \ \times 0.93) \times (0.41 \ \times 0.93) = 0.17$$

$MICICMcc[2,2] += (0.51\ \times 0.93) \times (0.59\ \times 0.93) = 0.26$

$MICICMcc[2,3] += (0.51\ \times 0.93) \times (0.41\ \times 0.93) = 0.18$

Figure 33 gives the matrix obtained with our method (MICICMcc) as well as that obtained by the method of ref.(Vadivel, Sural et al. 2007) (ICICM). From this figure, we see that ICICM (*H1*, *H2*) = 0 which means that, in our image, there are no red pixels followed by yellow ones, whereas, MICICMcc (*H1*, *H2*) = 0.43 which means such a situation exists. CCM has considered that *H2* never occurs after *H1* because it did not map any pixel to *H1*, whereas our MICICM considers pixels near the boundaries of *H1* as belonging to *H1* too. The two occurrences of (*H1*, *H2*) are *(I(1,1), I(1,2))* and *(I(3,2), I(3,3)).*

	H1	H2	H3	H4	H5	H6
H1	0	,46	0,38	0	0	0
H2	0,45	1,01	0,40	0	0	0
H3	0,36	0,41	0	0,19	0,17	0
H4	0	0	0,22	0,51	0,24	0
H5	0	0	0,20	0,21	0	0
H6	0	0,46	0,38	0	0	0

(a)

	H1	H2	H3	H4	H5	H6
H1	0	0	0	0	0	0
H2	0	3.80	0	0	0	0
H3	0	0	0	0	0	0
H4	0	0	0	1.90	0	0
H5	0	0	0	0	0	0
H6	0	0	0	0	0	0

(b)

Figure 33. The result of extracting the co-occurrence matrix. (a) MICICM based on our mapping scheme. (b) ICICM based on the standard hard mapping.

Finally, we extract four features namely, contrast, correlation, energy and homogeneity, from each matrix. Table 1 shows the results. We see that the features extracted from the CCM are:

- Contrast = 0: which suggests that the image is constant,
- Homogeneity = 1: which says that the image is absolutely homogeneous (or uniform) without any texture.

This is not precise at all since our image (Figure 29) was not constant, instead, its colors vary from pixel to pixel. As for the features calculated from our MICICM, they hold more meaningful information with Contrast = 1.3 and Homogeneity = 0.6.

Table 1. Features extracted from both CCM and MICICM.

	MICICM	ICICM
Contrast	1.3	0
Energy	0.1	0.5
Correlation	0.5	1
Homogeneity	0.6	1

V.5 Conclusion

In this chapter, we introduced our proposed generalization of GLCM in order to incorporate information about the colors within images. To do so, we firstly convert the image from RGB to HSV color space to eliminate the correlation between the channels. After sampling the color (resp., gray-level space), we perform a mapping task based on a set of weight assignment functions we propose. We then define the algorithm that lists the different steps of extracting the MICICM. Finally, we add up a detailed example of extracting the MICICM from a given image to make things clearer. In this later, we saw how our proposed method better deals with color information that the image incorporates by providing more meaningful features than the others does.

Chapter VI. EXPERIMENTATION AND VALIDATION

We have evaluated our MICICM in the context of content-based image retrieval, however, it can be used in different other applications of image processing and pattern recognition. In the following, we will start by explaining the experimental setup. Then, we will report the obtained results and discuss them.

VI.1 Experimental setup

As conditions of the experiment, we set the following:

VI.1.1 Dataset

numerous texture-dataset benchmarks have been proposed in literature such as Brodatz(Valkealahti and Oja 1998), Meastex and Vistex(Singh and Sharma 2001), Outex(Ojala, Maenpaa et al. 2002), etc.

In our research, we use OuTex benchmark(Ojala, Maenpaa et al. 2002) to perform our tests. There exists several releases of this dataset (The Center for Machine Vision and Signal Analysis (CMVS)). We choose Outex_TC_00016 because it contains the highest number of categories (29 categories, e.g., barley rice, canvas, chips, etc.) compared to the others. However, Outex_TC_00016 is made up of gray-level images only. As we want to test the color integration in the GLCM, color images are required. The color images that correspond to the ones of Outex_TC_00016 are available in(The Center for Machine Vision and Signal Analysis (CMVS)), we collect those images and we use them in our experiments. Figure 34 shows a representative sample from the collected images.

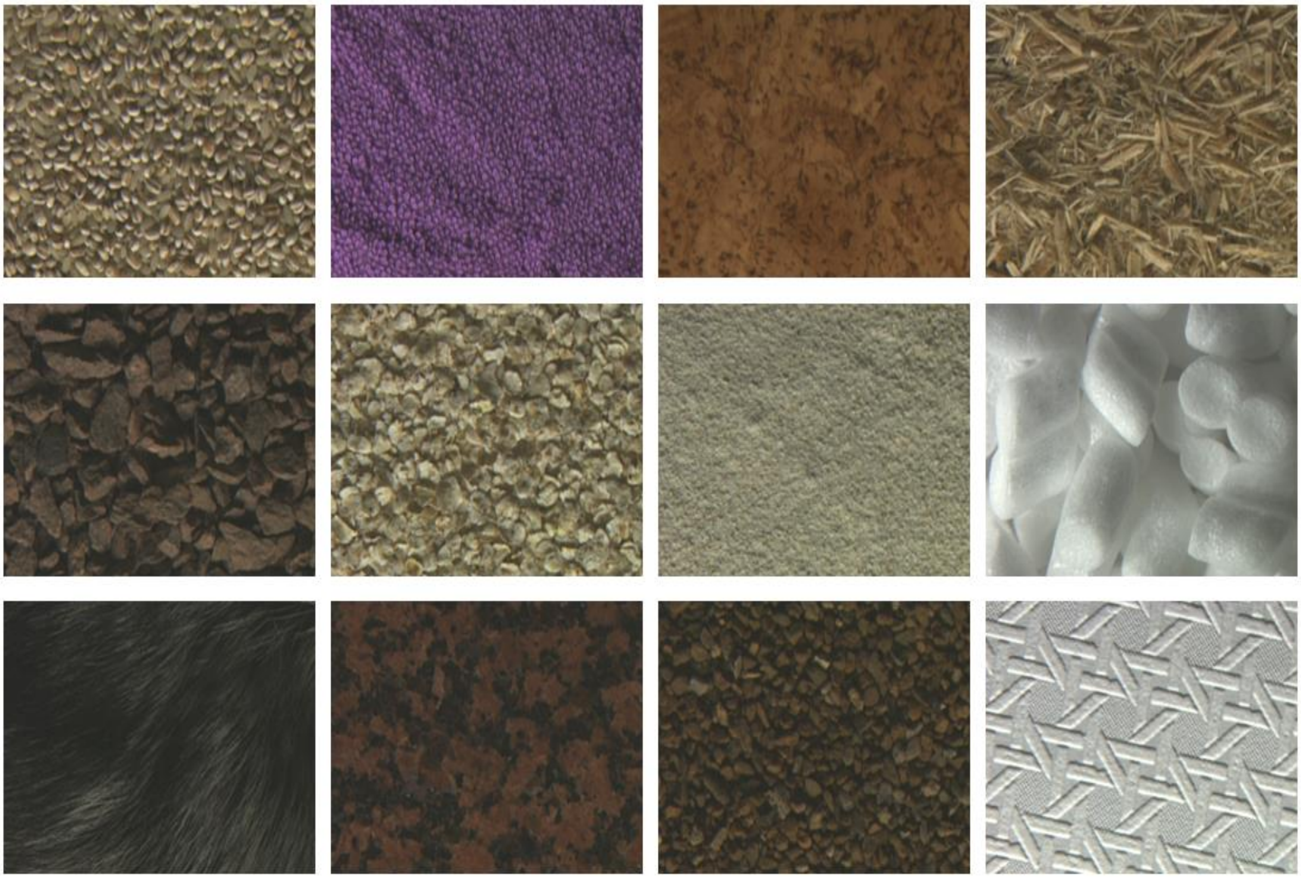


Figure 34. Representative sample from collected OuTex texture images.

- Each image has been rotated by 9 different angles (0^o, 5^o, 10^o, 15^o, 30^o, 45^o, 60^o, 75^o and 90^o) then subdivided into four equal sub-images which gives a total of 11484 images $(319 \times 9 \times 4)$, half of which (5742 images) has been used for indexing and the other half for retrieval. Our built dataset is accessible and freely available in (Khaldi and Mohammed Lamine, 2016).

VI.1.2 Texture features

- We compared the retrieval results of our proposed method MICICM (where hue and intensity channel are sampled, respectively, into 6 and 4 bins) with three other methods based on co-occurrence matrix: CCM(Palm 2004) (where hue channel is sampled into 6 bins), FCCM(Ledoux, Losson et al. 2015) (where, R, G, and B are sampled into 2, 4 and 2, respectively) and ICICM(Vadivel, Sural et al. 2007) (where, hue and intensity channel are sampled, respectively, into 6 and 4 bins).
- From each of these matrices, we extract four features, namely, contrast, correlation, energy and homogeneity according to Eq. (65)-(68).

$$Contrast = \sum_{i=1}^{N}\sum_{j=1}^{N}(i-j)^2 \,.\, C[i][j] \tag{65}$$

$$Correlation = \sum_{\boldsymbol{i=1}}^{\boldsymbol{N}}\sum_{\boldsymbol{j=1}}^{\boldsymbol{N}}(i-\ \mu).(j-\ \mu)\,.\,C[i][j] \tag{66}$$

$$Energy = \sum_{\boldsymbol{i=1}}^{\boldsymbol{N}}\sum_{\boldsymbol{j=1}}^{\boldsymbol{N}} C[i][j]^2 \tag{67}$$

$$Homogenteiy = \sum_{\boldsymbol{i=1}}^{\boldsymbol{N}}\sum_{\boldsymbol{j=1}}^{\boldsymbol{N}}\frac{1}{1+\ \ (i-j)^2}\,.\,C[i][j] \tag{68}$$

such that C is the co-occurrence matrix and μ is the mean which is calculated according to Eq. (69).

$$\mu = \sum_{i}\sum_{j} i\,.\,C[i][j] \tag{69}$$

VI.1.3 Evaluation metrics

Two main factors contribute to the quality of the retrieval. The first one is the percentage of retrieved relevant images, while the second is the right ranking of these latter. In order to evaluate the first aspect, we opted for two measures which have frequently been used in image retrieval: precision *(P)* and recall *(R)*:

- $P = \frac{Number\ of\ retrieved\ relevant\ images}{Total\ number\ of\ retrieved\ images}$
- $R = \frac{Number\ of\ retrieved\ relevant\ images}{Total\ number\ of\ relevant\ images\ in\ the\ dataset}$

In retrieval, a feature that yields well-ranked images with less precision is better than the one that yields bad-ranked images with higher precision. Figure 35 shows an example of such a case. It appears that retrieval result Figure 35.(b) is more accurate, although, retrieval result Figure 35.(a) has more relevant images.

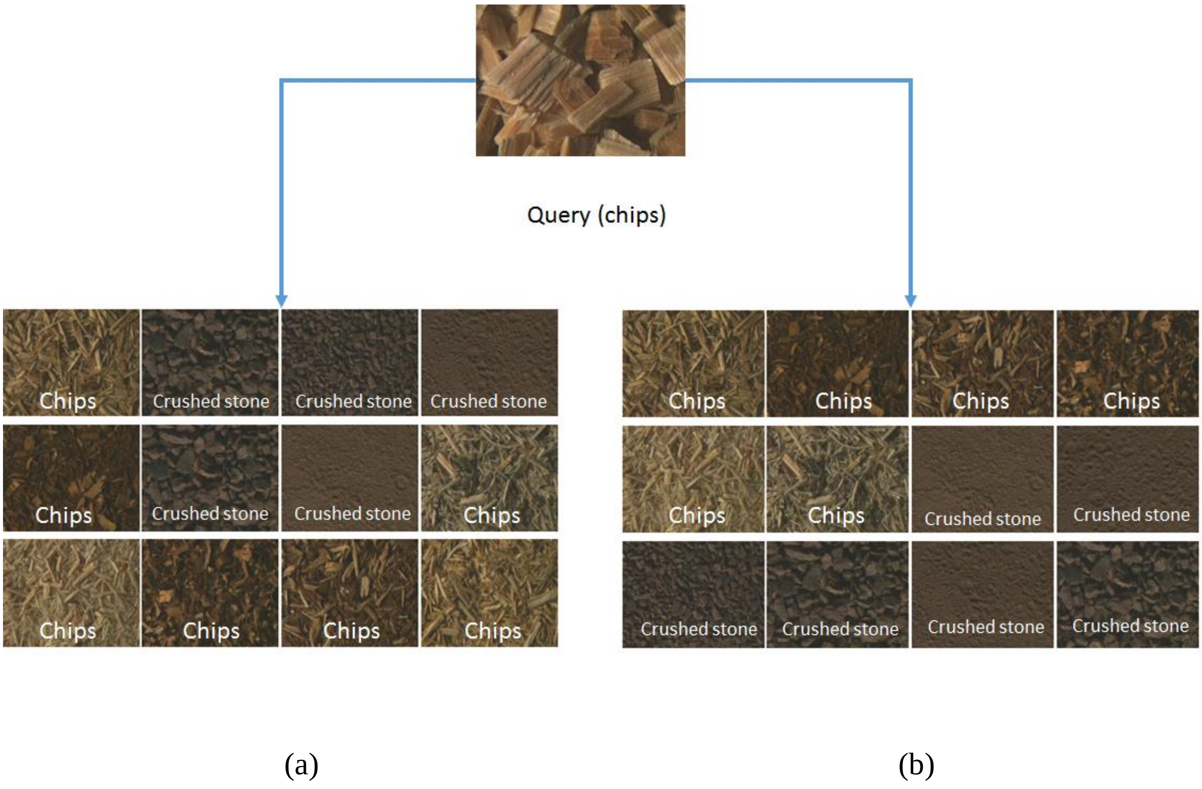


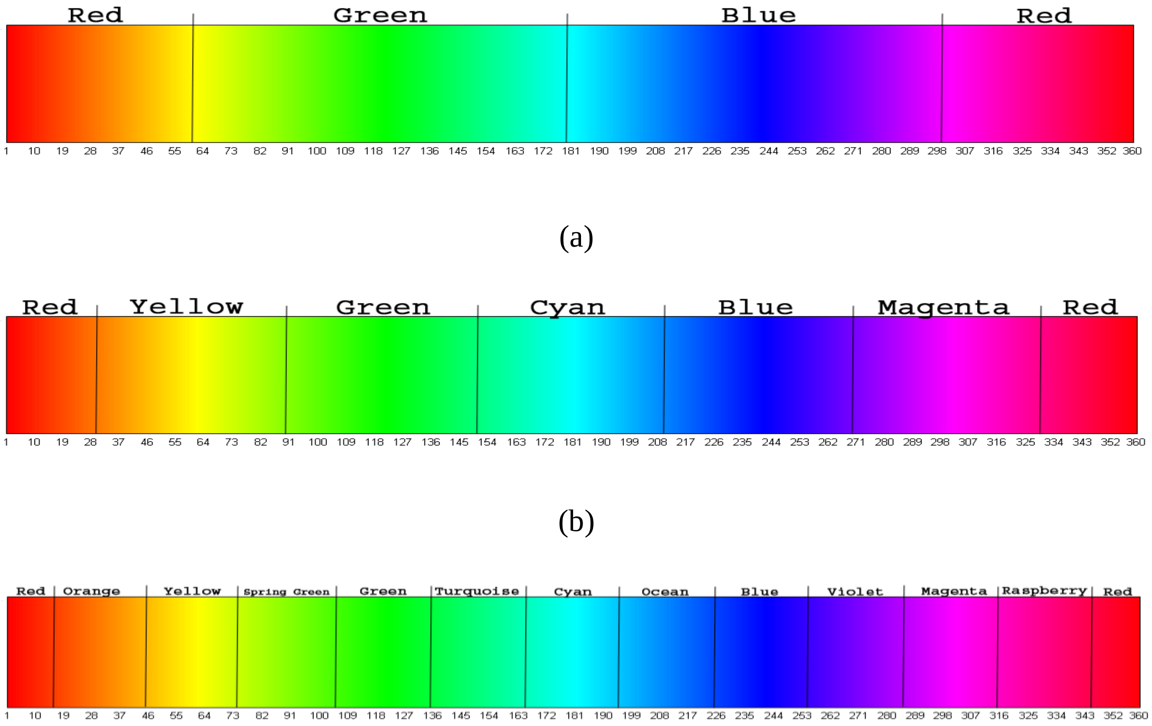
Figure 35. Retrieval results of two different features. Retrieval result (a) is less accurate than retrieval result (b) although it has more images that are pertinent.

In Such cases, the precision became a deceiver metric. To avoid falling in this problem, we resort to Mean Average Precision (*MAP*) which takes into account the ranking of the retrieved images. It is given by the following formula:

$MAP = \frac{\sum_{q=1}^{Q} AvgP(q)}{Q}$, Q is the number of queries and *AvgP(q)* is the average precision of a query q.

VI.1.4 HSV color space sampling

We quantize intensity and hue channels, respectively, into 4 and 6 bins. However, these channels can be quantized using more or less bins. Too much bins (Figure 36.(c)) means more calculation, whereas, too less (Figure 36.(a)) means miss-mapping. Figure 36 shows examples of hue channel quantizing including our used one (Figure 36.(b)). Anyway, reader should understand that channel quantization does not have to be systematic (equal bin sizes). Instead, it might be customized based on the application field.



(a)

(b)

(c)

Figure 36. Examples of hue channel quantizing: (a) quantized to 3 bins, (b) quantized to 6 bins and (c) quantized to 12 bins.

VI.2 Experimental results

In order to prove the efficiency of our method, we carried out three experiments, each is intended to measure a specific aspect. In the first experiment, we determine the best values of the parameters *t1* and *t2* for Eq. (61); *k1* and *k2* for Eq. (63). The aim of the second experiment is to, separately, compare features extracted from our MICICM with those extracted from CCM, FCCM and ICICM. The third experiment is dedicated to compare all possible combinations of features extracted using each of the three methods. We report results in terms of precision, recall and MAP.

VI.2.1 Choice of parameters

In this experiment, we try to determine the best values of the parameters *t1*, *t2*, *k1* and *k2* of the equations Eq.(61)-(64). In order to estimate these parameters correctly, we sample 12 images from each category of texture; given that the total number of categories is 29, we obtain a sub-set of 348 images. The half of this sub-set has been used for indexation and the other half for retrieval. By considering a recall equal to 1, retrieval process is carried out using different combinations of the parameters $t1, t2, k1$ *and* $k2$ .The yielded precisions for each combination are averaged then compared with the others. The best values of the parameters correspond to the combination that yields the highest average precision. Figure 37 shows examples of Eq.(61) plotted using different values of $t1$ and $t2$.

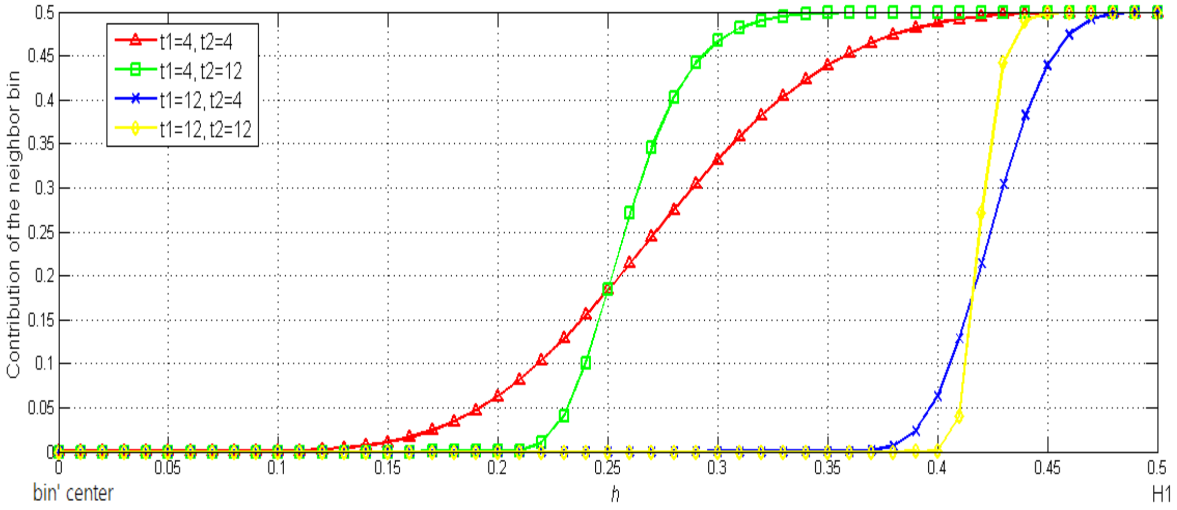


Figure 37. The contribution of a neighbor bin to a given hue value *h* according to the distance between *h* and the boundary *H1*. We use different value combinations for the parameters *t1* and *t2*.

From Figure 37 we see that the higher values of parameters are, the less neighbor contribution is. For example, for *t1*= 12 and *t2* =12 the neighbor-bin contribution begins if the hue or the gray –level value is greater than 0.4. However, for *t1*= 4 and *t2*= 4 the neighbor bin starts contributing to the color or gray level if its value is greater than 0.1. We set $t1 > 4$, otherwise, the neighbor contribution will be extended to reach pure colors which is wrong. In contrast, if it is too high, it will yield a hard mapping.

Firstly, we set *k1*, *k2* and *t2* to 4 then estimate the precision for different values of *t1*. The obtained results using different values of *t1* are listed in Table 2.

Table 2. Precision yielded using Eq.(61) by setting *k1*, *k2* and *t2* to 4, and $t1 > 4$.

t1	4	5	6	7	8	9	10	11	12
precision	0,130	0,128	0,132	0,136	0,146	0,147	0,143	0,142	0,140

From **Table 2**, we see that the precision increments until it reaches the maximum of 0.147. This maximum is yielded when we assign the value 9 to the parameter *t1*.

In the same way, the parameter *t2* is estimated by setting *k1*, *k2* to 4 and *t1* to its best value (i.e., 9). The obtained results are listed in Table 3.

Table 3. Precision yielded using Eq.(61) by setting *k1* and *k2* to 4, *t1* to 9, and $t2 > 4$.

t2	4	5	6	7	8	9	10	11	12
precision	0,1471	0,1480	0,1493	0,1490	0,1484	0,1486	0,1483	0,1488	0,1487

From **Table 3**, we see that the precision increments until it reaches the maximum of 0.1493. This maximum is yielded when we assign the value 6 to the parameter *t2*.

From the two previous evaluations, we have considered that the best results will be obtained when the values of the parameters *t1* and *t2* are respectively 9 and 6. In Figure 38 , Eq. (61) is plotted using these two values (i.e., *t1= 9* and *t2= 6)*. This curve seems to be satisfying to our constraints a-d.

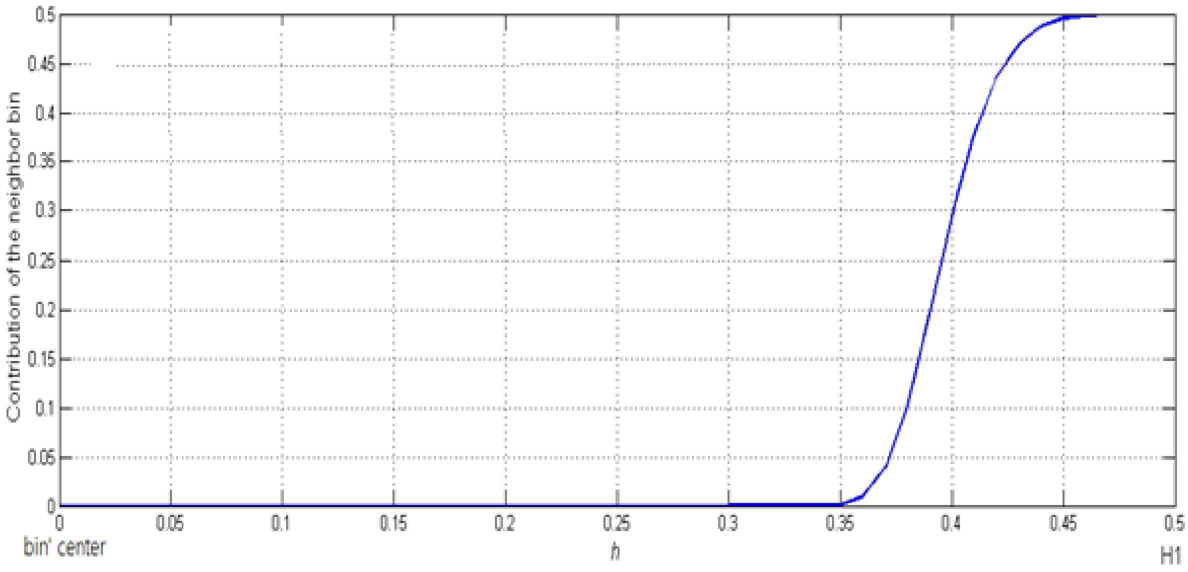


Figure 38. Eq. (61) plotted by setting the parameters *t1* = 9 and *t2* = 6.

Similarly, to define which values are best for the parameters *k1* and *k2* of Eq. (63), we perform the same steps as we did with *t1* and *t2*. Table 4 shows the results of evaluating *k1* by setting *t1*, *t2* and *k2* to 4;

Table 4. Precision yielded using Eq. **(63)** by setting *t1*, *t2* and *k2* to 4, and $k1 > 4$.

k1	4	5	6	7	8	9	10	11	12	13	14	15
precision	0,140	0,138	0,131	0,139	0,143	0,144	0,143	0,144	0,145	0,140	0,140	0,139

From **Table 4**, we see that the precision increments until it reaches the maximum of 0.145. This maximum is yielded when *k1* is equal to *12*.

In the same way, the parameters *k2* is estimated by setting *t1*, *t2* to 4 and *k1* to 9, results are listed in Table 5.

Table 5. Precision yielded using Eq**. (63)** by setting *t1* and *t2* to 4, *k1* to 12, and $k2 > 4$.

k2	4	5	6	7	8	9	10	11	12
precision	0,1455	0,1458	0,1447	0,1448	0,1446	0,1447	0,1449	0,1449	0,1449

From Table 5, we see that the precision increments until it reaches the maximum of 0.1458. This maximum is yielded when *k2* is equal to *5*.

VI.2.1.1 Conducting a Second Iteration

In order to assure that these values are the best, we conduct a second iteration. Each time, three of these parameters are set to their best values, and then we estimate the precision while changing the fourth one. The obtained results are illustrated in Table 6.

Table 6. Results of a second iteration for estimating *t1*, *t2*, *k1* and *k2*.

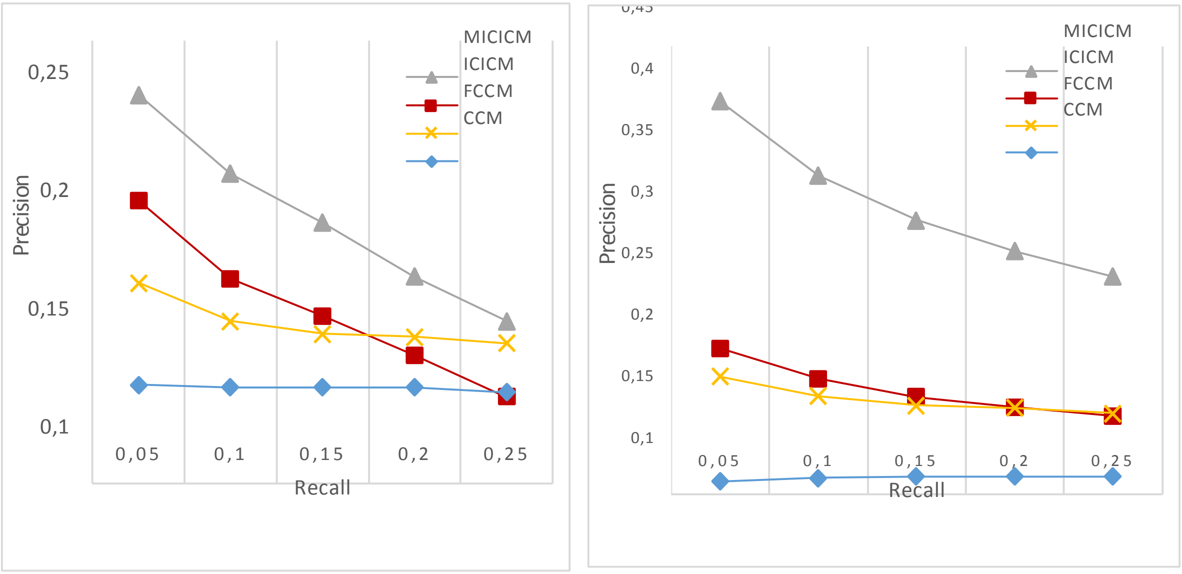
	4	5	6	7	8	9	10	11	12
t1	0,1545	0,1519	0,1522	0,1539	0,1550	0,1558	0,1541	0,1440	0,1430
t2	0,1378	0,1492	0,1558	0,1512	0,1509	0,1396	0,1389	0,1391	0,1388
k1	0,1418	0,1387	0,1411	0,1495	0,1474	0,1482	0,1508	0,1522	0,1558

k2	0,1518	0,1558	0,1511	0,1495	0,1474	0,1482	0,1488	0,1479	0,1496

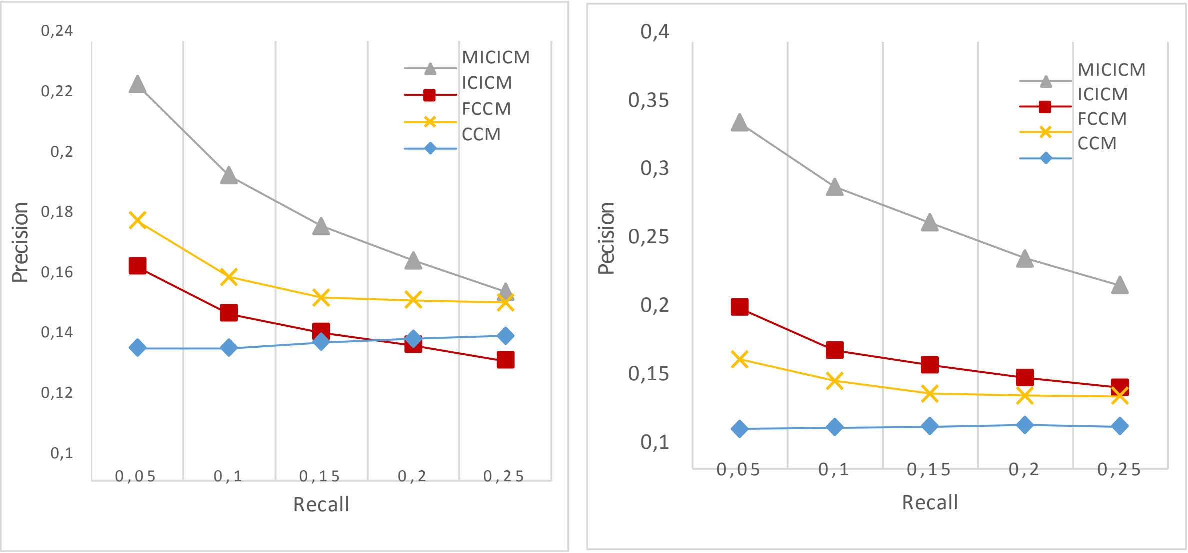
From the previous estimations, we found out that our method yields the best precisions when the values of the parameters *t1*, *t2*, *k1* and *k2* respectively are 9, 6, 12 and 5. By considering these values, we conduct evaluations on the full dataset in the following two subsections.

VI.2.2 Individual Features Evaluation

After having features extracted (i.e., contrast, correlation, energy and homogeneity) from MICICM, FCCM, CCM and ICICM, we conduct retrieval using each feature separately. We take each image, from those dedicated to retrieval, as a query, and then we average recalls and precisions yielded by all the queries. The obtained results are shown in Figure 39.



(a) (b)



(c) (d)

Figure 39. Features Evaluation. (a) Contrast, (b) Correlation, (c) Energy and (d) Homogeneity.

From Figure 39, it appears that our method significantly outperforms the FCCM, CCM and ICICM for all features. In addition, as the recall increases, our method still outperform the three others. By taking homogeneity feature (i.e., Figure 39.(d)) as an example, we notice that the CCM has yielded the lowest result. This is may be explained by the fact that some images are homogeneous in terms of color, however, the shade of colors may differs from pixel to another which is undetectable by capturing color information only. ICICM has overcome this drawback by considering both color and intensity, and yielded better performance than FCCM and CCM, whereas our MICICM has outperformed the others. By adopting the smart mapping, which is not taken into account by ICICM, our MICICM has successfully dealt with the images in which colors are near the boundaries. This confirms the hypothesis arguing that using hard mapping degrades retrieval results.

VI.2.3 Combined Features Evaluation

Since that in the previous experiment we have considered each feature separately, the aim of the current experiment is to prove that combining these features does not degrade the retrieval performance but rather improves it. The quadruple *(i, j, k, l)*, which correspond respectively to contrast, correlation, energy and homogeneity, represents the features that have been taken into consideration. A value *1* in a quadruple means that the corresponding feature is considered, *0* in the opposite case. For example, *1101* refers to a combination of contrast, correlation and homogeneity. Figure 40 illustrates the obtained results for each combination.

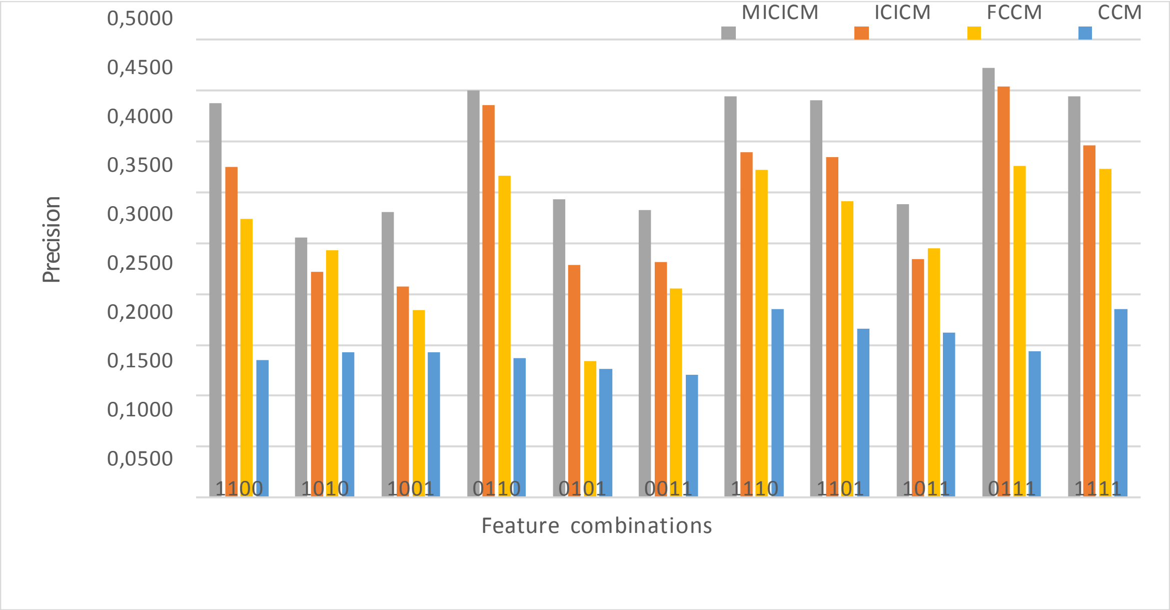


Figure 40. Precision histogram obtained using all possible feature combinations. The quadruple (*i*, *j*, *k*, *l*) corresponds, respectively, to contrast, correlation, energy and homogeneity (e.g., *1101* refers to a combination of contrast, correlation and homogeneity).

From Figure 40 we notice that our method outperforms the ICICM, FCCM and CCM in all the possible combinations by an average of *5%*, *9%* and *21%*, respectively.

In retrieval, a feature that yields well-ranked images with less precision is better than the one that yields bad-ranked images with higher precision. In such cases, the precision became a deceiver metric. To avoid falling in this problem, we employ the *MAP* metric. We have re-evaluated all the possible combination of features with MAP metric. The obtained results have been illustrated in Figure 41.

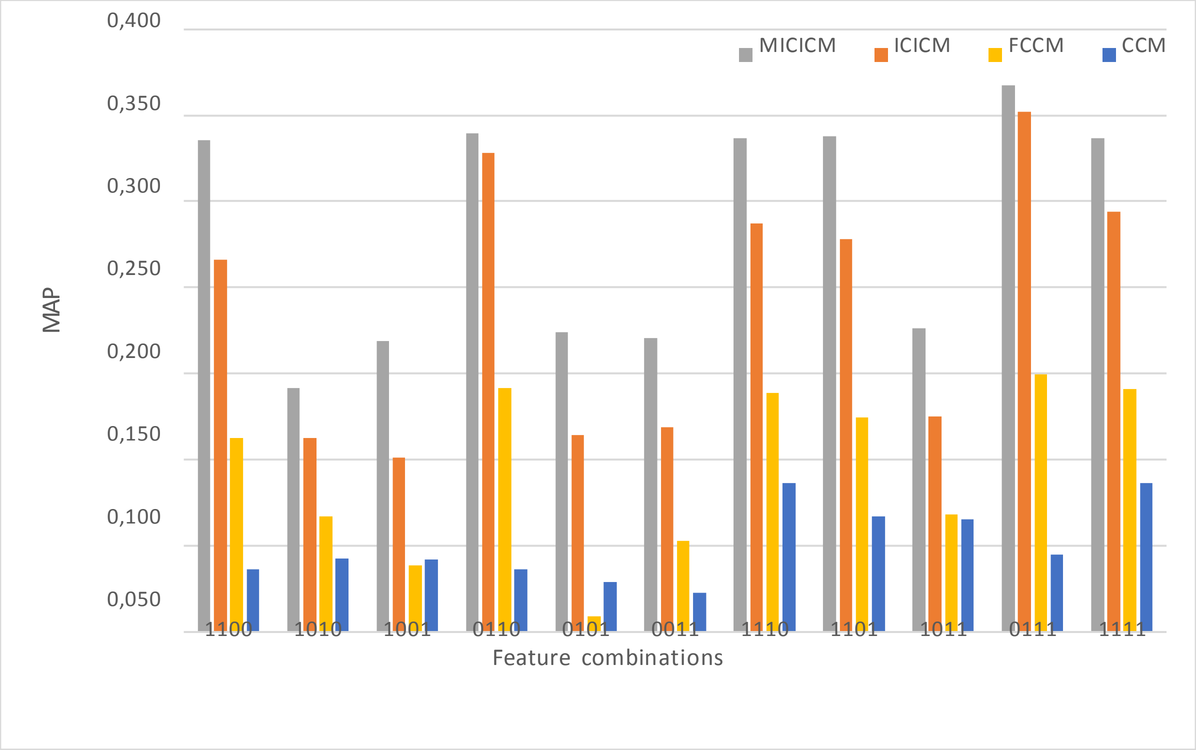


Figure 41. Histogram of mean average precession (MAP) of CCM, ICICM and MICICM. The quadruple (*i*, *j*, *k*, *l*) corresponds, respectively, to contrast, correlation, energy and homogeneity (e.g., *1101* refers to a combination of contrast, correlation and homogeneity).

According to this Figure 41, our MICICM brings more relevant images in top positions than the two other methods. This has been achieved thanks to our mapping scheme where colors and gray levels are mapped in a better way.

Let us now take a look at the behavior of MAP for the four methods when the recall increases. We consider the combination of features *1100* where the results are shown in Figure 42. We see that the difference between our method and the three others is noticeable. This proves that even for higher values of recall, the MAP achieved by our method still exceed the other ones.

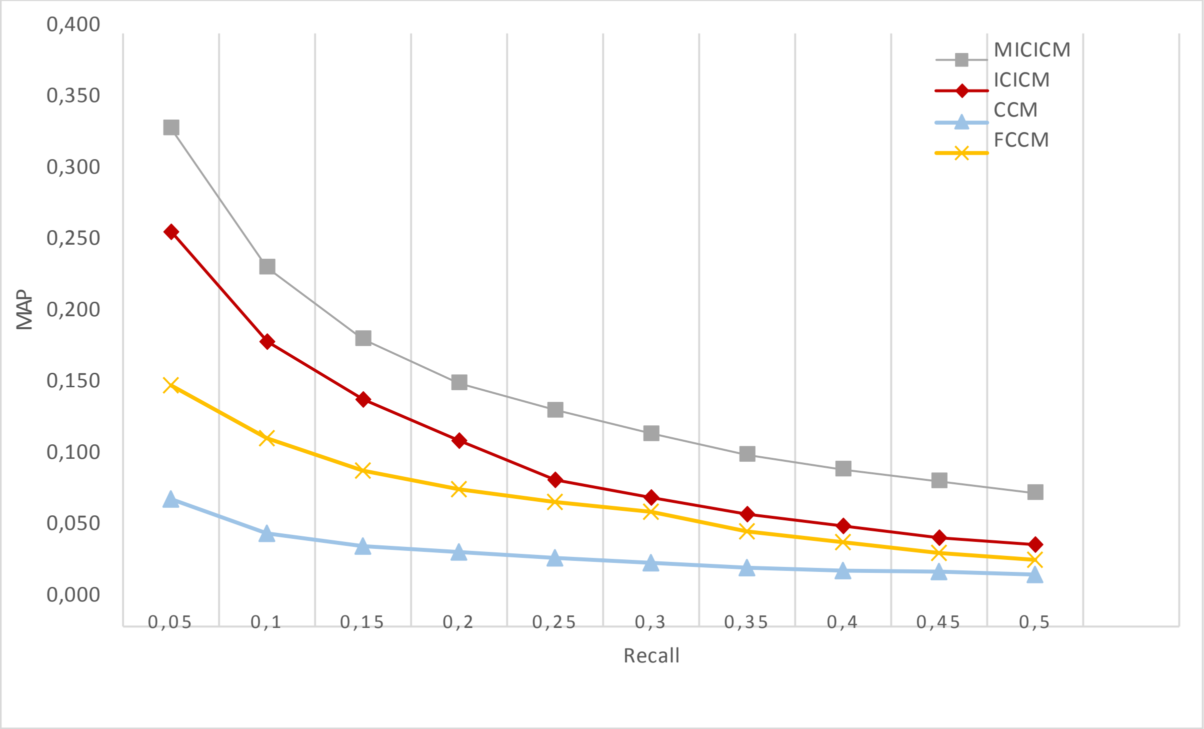


Figure 42. MAP behavior in terms of recall $R\epsilon[0{,}05\ 0{,}5]$.

From Figure 42, we notice that the three methods CCM, FCCM and ICICM are converging to the same MAP value while the recall increases. In addition, we see that when the recall reaches the value 0.5 the difference in MAP between these three methods become trivial. At the opposite, this difference remains high between the MAP yielded by our method and the ones of the others. This difference is not affected too much even if the recall increases.

VI.3 Conclusion

In this chapter, we have evaluated the performance of our *MICICM* against three other related works namely *CCM*, *ICICM* and *FCCM*. This evaluation has been done considering three aspects that are precision, recall and map. Due to the high correlation between the 14 haralick's features, we took into account only the four least correlated features namely contrast, correlation, variance and homogeneity. As a dataset, we built and made online available a color dataset based on a gray-level Outex_TC_00016 dataset which has the highest number of categories and classes. After estimating the parameters *t1, t2, k1* and *k2*, and conducting the experiments, the results indicates that our *MICICM* has remarkably outperformed the others in terms of single and combined features.

GENERAL CONCLUSION

Digital image analysis is the art of using the techniques that allow a machine (i.e., computer or some other electrical device) to, automatically, study an image in order to obtain useful information about its content. These information could be, then, exploited to take a decision by a specialist or the machine itself.

In machine learning, pattern recognition and image processing, a feature could be defined as a vector of values that are derived from an initial big set of measured data (i.e., dimensionality reduction). Feature vectors generally tend to be informative and non-redundant. A specific structure in the image can often be represented in different ways based on the selected image feature. Image features can be broadly categorized into four main categories namely, color, texture interest point and shape features.

GLCM is amongst the well-known and widely used techniques for texture representation. This matrix accumulates a set of second order statistics that measures the spatial dependency of two gray levels given a displacement vector $(\Delta x, \Delta y)$. Although the GLCM has proven its effectiveness as a texture representation, four main issues should be addressed in order to improve its accuracy and the results it yields:

a. *Color integration:* Since the GLCM is computed from gray-level images only, color information, which may be very important, is totally ignored.
b. *Multi-spectral integration:* integrate information extracted from the different spectral bands, composing a multi-spectral image, into a co-occurrence matrix.
c. *Color sampling and mapping:* Color sampling and mapping are critical processes since the better they are, the better results are.
d. *Extracting new features:* Besides the features introduced by Haralick in his original work on GLCM, many other authors have proposed new ones.

In this research, we propose a generalization of GLCM so it can be used with color texture images without resort to any other color features. This representation which we refer to as Modified Integrative Color Intensity Co-occurrence Matrix (MICICM) holds information about both color and intensity. To reach such a generalization, a mapping function, which determine for each pixel value the bin it falls into, is needed. In the previous studies, this function uses a hard mapping where all pixel values that fall in a bin are considered as the same, regardless their values. This presents a number of drawbacks. To fix them, we introduce a new color and gray-level mapping scheme based on a set of weight assignment functions we propose. In our scheme, each pixel is mapped to more than one possible color (and gray level) bin, in order to avoid the drawbacks of hard mapping. This scheme is based on a set of weight assignment functions that we define, where each pixel is mapped to more than one possible color (and gray-level) bin, in order to avoid the drawbacks of hard color mapping. This is justified by the fact that most hue values in the Hue, Saturation, and Value (HSV) color space result from the mixing of more than one color. By using the MICICM, the image is compactly represented with few features that come in form of statistical moments, which make it suitable for real time systems. It can be used for different applications of image processing and pattern recognition.

We have evaluated our MICICM in the context of content-based image retrieval, however, it can be used in different other applications of image processing and pattern recognition. Experimental results show that our method significantly outperforms others of the literature in terms of precision, recall and MAP.

The findings of this investigation complement those of earlier studies and make several noteworthy contributions to image representation. In the future, we will adopt the same principle in order to improve other existing features. Moreover, the proposed integrative feature can be used, in future works, to fulfil other tasks such as texture recognition.

Bibliography

Aiadi, O., B. Khaldi and M. L. Kherfi (2016). "An Automated System for Date Fruit Recognition Through Images " Revue des bio ressources **6**(1).

Aiadi, O. and M. L. Kherfi (2016). "A new method for automatic date fruit classification." International Journal of Computational Vision and Robotics (IJCVR).

Alahi, A., R. Ortiz and P. Vandergheynst (2012). Freak: Fast retina keypoint. International Conference on Computer Vision and Pattern Recognition (CVPR 2012), Providence, RI, USA, IEEE.

Arvis, V., C. Debain, M. Berducat and A. Benassi (2011). "Generalization of the cooccurrence matrix for colour images: application to colour texture classification." Image Analysis & Stereology **23**(1): 63-72.

Bandhyopadhyay, D. S. K. and T. U. Paul (2012). "Segmentation of brain MRI image–a review." International Journal of Advanced Research in Computer Science and Software Engineering **2**(3): 409-413.

Barnard, K. and D. Forsyth (2001). Learning the semantics of words and pictures. in Proceedings of the Eighth International Conference on Computer Vision (ICCV 2001), Vancouver, British Columbia, Canada, IEEE.

Bay, H., A. Ess, T. Tuytelaars and L. Van Gool (2008). "Speeded-up robust features (SURF)." Computer Vision and Image Understanding, Elsevier **110**(3): 346-359.

Bezdek, J. C. (2013). Pattern recognition with fuzzy objective function algorithms, Springer Science & Business Media.

Bijhold, J. and Z. Geradts (2002). "Content based information retrieval in forensic image databases." Journal of Forensic Science **47**(2): 285-292.

Bird, C., P. Elliott and E. Griffiths (1996). User interfaces for content-based image retrieval. Colloquium on Intelligent Image Databases, IET.

Borst, A. and M. Egelhaaf (1989). "Principles of visual motion detection." Trends in neurosciences **12**(8): 297-306.

Calonder, M., V. Lepetit and P. Fua (2008). Keypoint signatures for fast learning and recognition. European Conference on Computer Vision (ECCV 2008). D. Forsyth, P. Torr and A. Zisserman. Marseille, France, Springer Berlin Heidelberg**:** 58-71.

Calonder, M., V. Lepetit, C. Strecha and P. Fua (2010). Brief: Binary robust independent elementary features. European conference on computer vision (ECCV 2010), Crete, Greece from, Springer Berlin Heidelberg.

Canny, J. (1986). "A computational approach to edge detection." Transactions on Pattern Analysis and Machine Intelligence (PAMI 1986), IEEE **8**(6): 679-698.

Cao, L.-L., X.-L. Li, N.-H. Yu and Z.-K. Liu (2002). Naked people retrieval based on adaboost learning. Proceedings of the International Conference on Machine Learning and Cybernetics, Beijing, CHINA, IEEE.

Çarkacıoğlu, A. and F. Yarman-Vural (2003). "SASI: a generic texture descriptor for image retrieval." Pattern Recognition, Elsevier **36**(11): 2615-2633.

Carkacioglu, A. and F. Yarman-Yural (2001). SASI: a new texture descriptor for content based image retrieval. Proceedings of the International Conference on Image Processing (ICIP 2001), Thessaloniki, Greece, IEEE.

Carloni, R., V. Lippiello, M. D'auria, M. Fumagalli, A. Y. Mersha, S. Stramigioli and B. Siciliano (2013). "Robot vision: Obstacle-avoidance techniques for unmanned aerial vehicles." Robotics & Automation Magazine, IEEE **20**(4): 22-31.

Cevikalp, H. and B. Triggs (2010). Face recognition based on image sets. Conference on Computer Vision and Pattern Recognition (CVPR 2010), San Francisco, CA, USA, IEEE.

Chaira, T. and A. Ray (2005). "Fuzzy measures for color image retrieval." Fuzzy Sets and Systems **150**(3): 545-560.

Chan, Y., R. Harvey and D. Smith (1999). Building systems to block pornography. BCS Electronic Workshops in Challenge of Image Retrieval.

Chang, E. Y., J. Z. Wang, C. Li and G. Wiederhold (1998). Rime: A replicated image detector for the world wide web. Photonics East (ISAM, VVDC, IEMB), Boston, Massachusetts, International Society for Optics and Photonics.

Chang, S.-F., T. Sikora and A. Purl (2001). "Overview of the MPEG-7 standard." IEEE Transactions on circuits and systems for video technology **11**(6): 688-695.

Chang, Y., Q. Zaman, A. Schumann, D. Percival, T. Esau and G. Ayalew (2012). "Development of color co-occurrence matrix based machine vision algorithms for wild blueberry fields." Applied Engineering in Agriculture **28**(3): 315-323.

Chen, C.-h., L.-F. Pau and P. S.-p. Wang (1993). Handbook of pattern recognition and computer vision, World Scientific.

Chen, C. and J. Z. Wang (2002). Large-scale Emperor Digital Library and semantics-sensitive region-based retrieval. The Proceedings of Digital Library—IT Opportunities and Challenges in the New Millennium, Beijing, China.

Chen, J., S. Shan, C. He, G. Zhao, M. Pietikainen, X. Chen and W. Gao (2010). "WLD: A robust local image descriptor." Transactions on Pattern Analysis and Machine Intelligence (PAMI 2010), IEEE **32**(9): 1705-1720.

Chen, J., S. Shan, G. Zhao, X. Chen, W. Gao and M. Pietikainen (2008). A robust descriptor based on weber's law. Conference on Computer Vision and Pattern Recognition (CVPR 2008), Anchorage, Alaska, USA, IEEE.

Chen, X., K. Jia, X. Lian and S. Wei (2012). Color Image Retrieval Based on Color and Texture Features. Advances in Computer Science and Information Engineering, Springer**:** 669-674.

Cross, G. R. and A. K. Jain (1983). "Markov random field texture models." Transactions on Pattern Analysis and Machine Intelligence (PAMI 1983), IEEE(1): 25-39.

Dacheng, T., L. Xuelong, Y. Yuan, Y. Nenghai, L. Zhengkai and T. Xiao-ou (2002). A set of novel textural features based on 3D co-occurrence matrix for content-based image retrieval. Proceedings of the Fifth International Conference on Information Fusion (FUSION 2002) Annapolis, Maryland, USA, IEEE.

Dalal, N. and B. Triggs (2005). Histograms of oriented gradients for human detection. Conference on Computer Vision and Pattern Recognition (CVPR 2005), San Diego, CA, USA, IEEE.

Daugman, J. G. (1980). "Two-dimensional spectral analysis of cortical receptive field profiles." Vision research **20**(10): 847-856.

Daugman, J. G. (1985). "Uncertainty relation for resolution in space, spatial frequency, and orientation optimized by two-dimensional visual cortical filters." JOSA A **2**(7): 1160-1169.

Deng, Y., B. Manjunath, C. Kenney, M. S. Moore and H. Shin (2001). "An efficient color representation for image retrieval." Transactions on Image processing (TIP 2001), IEEE **10**(1): 140-147.

Deselaers, T., D. Keysers and H. Ney (2008). "Features for image retrieval: an experimental comparison." Information Retrieval **11**(2): 77-107.

Drimbarean, A. and P. F. Whelan (2001). "Experiments in colour texture analysis." Pattern recognition Letters, Elsevier **22**(10): 1161-1167.

Dunn, J. C. (1973). "A fuzzy relative of the ISODATA process and its use in detecting compact well-separated clusters."

Eakins, J. P. and M. E. Graham (1999). Content based image retrieval: A report to the JISC technology applications programme.

Fan, Z.-G., J. Li, B. Wu and Y. Wu (2008). Local patterns constrained image histograms for image retrieval. International Conference on Image Processing (ICIP 2008), San Diego, California, USA, IEEE.

Farrokhnia, F. and A. K. Jain (1991). A multi-channel filtering approach to texture segmentation. Conference on Computer Vision and Pattern Recognition (CVPR 1991), Lahaina, Maui, Hawaii, USA, IEEE.

Fleck, M. M., D. A. Forsyth and C. Bregler (1996). Finding naked people. European Conference on Computer Vision (ECCV 1996), Cambridge, UK, Springer.

Flickner, M., H. Sawhney, W. Niblack, J. Ashley, Q. Huang, B. Dom, M. Gorkani, J. Hafner, D. Lee and D. Petkovic (1995). "Query by image and video content: The QBIC system." Computer **28**(9): 23-32.

Galbally, J., S. Marcel and J. Fierrez (2014). "Image quality assessment for fake biometric detection: Application to iris, fingerprint, and face recognition." Transactions on Image Processing (TIP 2014), IEEE **23**(2): 710-724.

Gevers, T. and A. W. Smeulders (2000). "Pictoseek: Combining color and shape invariant features for image retrieval." Transactions on Image Processing (TIP 2000), IEEE **9**(1): 102-119.

Grimson, W. E. L. and D. P. Huttenlocher (1990). Object recognition by computer: the role of geometric constraints, Mit Press.

Gudivada, V. N. and V. V. Raghavan (1995). "Content based image retrieval systems." Computer **28**(9): 18-22.

Hadjidemetriou, E., M. D. Grossberg and S. K. Nayar (2004). "Multiresolution histograms and their use for recognition." Transactions on Pattern Analysis and Machine Intelligence (PAMI 2004), IEEE **26**(7): 831-847.

Han, J. and K.-K. Ma (2007). "Rotation-invariant and scale-invariant Gabor features for texture image retrieval." Image and Vision Computing, Elsevier **25**(9): 1474-1481.

Haralick, R. M. (1979). "Statistical and structural approaches to texture." Proceedings of the IEEE **67**(5): 786-804.

Haralick, R. M. and K. Shanmugam (1973). "Textural features for image classification." Transactions on Systems, Man, and Cybernetics, IEEE (6): 610-621.

Harder, N. and K. Rohr (2013). "Cell tracking and mitosis detection in 2D and 3D microscopy image sequences." The First Cell Tracking Challenge ,10th IEEE International Symposium on Biomedical Imaging (ISBI 2013).

Hauta-Kasari, M., J. Parkkinen, T. Jaaskelainen and R. Lenz (1996). Generalized co-occurrence matrix for multispectral texture analysis. Proceedings of International Conference on Pattern Recognition (ICPR 1996), Vienna, Austria, IEEE.

Hawkins, J. K. (1970). "Textural properties for pattern recognition." Picture processing and psychopictorics: 347-370.

Heckemann, R. A., S. Keihaninejad, P. Aljabar, D. Rueckert, J. V. Hajnal, A. Hammers and A. s. D. N. Initiative (2010). "Improving intersubject image registration using tissue-class information benefits robustness and accuracy of multi-atlas based anatomical segmentation." Neuroimage **51**(1): 221-227.

Huang, C.-B. and Q. Liu (2007). An orientation independent texture descriptor for image retrieval. International Conference on Communications, Circuits and Systems (ICCCAS 2007), Kokura, Japan, IEEE.

Huang, J., S. R. Kumar, M. Mitra, W.-J. Zhu and R. Zabih (1997). Image indexing using color correlograms. Proceedings of the Conference on Computer Vision and Pattern Recognition (CVPR 1997), San Juan, Puerto Rico, IEEE.

Huang, K.-Y. (2007). "Application of artificial neural network for detecting Phalaenopsis seedling diseases using color and texture features." Computers and Electronics in agriculture **57**(1): 3-11.

Huang, X., X. Liu and L. Zhang (2014). "A multichannel gray level co-occurrence matrix for multi/hyperspectral image texture representation." Remote Sensing **6**(9): 8424-8445.

Jain, A. K., S. Goel, S. Agarwal, V. Mittal, H. Sharma and R. Mahindru (1997). Multimedia systems for art and culture: A case study of Brihadisvara Temple. Procedings on Storage and Retrieval for Image and Video Databases, San Jose, CA, USA, SPIE.

Jain, R. (1995). "World-wide maze." MultiMedia, IEEE **2**(2): 3.

Jalaja, K., C. Bhagvati, B. L. Deekshatulu and A. K. Pujari (2005). Texture element feature characterizations for CBIR. Proceedings of the International Geoscience and Remote Sensing Symposium (IGARSS 2005), Seoul, Korea, IEEE.

Janney, P. and Z. Yu (2007). Invariant Features of Local Texturesa rotation invariant local texture descriptor. Conference on Computer Vision and Pattern Recognition (CVPR 2007), Minneapolis, Minnesota, USA, IEEE.

Julesz, B. (1981). "Textons, the elements of texture perception, and their interactions." Nature **290**(5802): 91-97.

Kavitha, C., D. B. P. Rao and D. A. Govardhan (2011). "Image retrieval based on color and texture features of the image sub-blocks." International Journal of Computer Applications (0975–8887) **15**(7).

Kaya, Y. and L. Kayci (2014). "Application of artificial neural network for automatic detection of butterfly species using color and texture features." The visual computer **30**(1): 71-79.

Khaldi, B. and M. L. Kherfi (2016). "Modified integrative color intensity co-occurrence matrix for texture image representation." Journal of Electronic Imaging (JEI), SPIE **25**(5).

Khaldi, B. and K. Mohammed Lamine. (29 July 2016). "Extended Outex texture retrieval test dataset." Retrieved 29 July 2016, from http://outex.comxa.com/.

Khelifi, R., M. Adel and S. Bourennane (2010). Texture classification for multi-spectral images using spatial and spectral gray level differences. International Conference on Image Processing Theory Tools and Applications (IPTA 2010), Paris, France, IEEE.

Kherfi, M. L. (2008). Review of human-computer interaction issues in image retrieval. Advances in Human Computer Interaction. P. Shane, INTECH Open Access Publisher**:** 215-240.

Kherfi, M. L., D. Ziou and A. Bernardi (2004). "Image retrieval from the world wide web: Issues, techniques, and systems." Computing Surveys (CSUR), ACM **36**(1): 35-67.

Kiranyaz, S., M. Ferreira and M. Gabbouj (2008). "A generic shape/texture descriptor over multiscale edge field: 2-D walking ant histogram." Transactions on Image Processing (TIP 2008), IEEE **17**(3): 377-391.

Kong, F.-H. (2009). Image retrieval using both color and texture features. 2009 International Conference on Machine Learning and Cybernetics, IEEE.

Kovalev, V. and S. Volmer (1998). Color co-occurrence descriptors for querying-by-example. Proceedings on Multimedia Modeling (MMM 1998), , Lausanne, Switzerland, Springer.

Kullback, S. (1987). "Letter to the editor: The Kullback-Leibler distance." American Statistician **41**(4): 340-341.

Laws, K. I. (1980). Textured image segmentation, DTIC Document, University of Southern California Los Angeles Image Processing INST.

Ledoux, A., O. Losson and L. Macaire (2015). Texture classification with fuzzy color co-occurrence matrices. International Conference on Image Processing (ICIP 2015), Quebec City, QC, Canada, IEEE.

Lee, H. Y., H. K. Lee and Y. H. Ha (2003). "Spatial color descriptor for image retrieval and video segmentation." Transactions on Multimedia, IEEE **5**(3): 358-367.

Lee, K.-L. and L.-H. Chen (2005). "An efficient computation method for the texture browsing descriptor of MPEG-7." Image and Vision Computing, Elsevier **23**(5): 479-489.

Leutenegger, S., M. Chli and R. Y. Siegwart (2011). BRISK: Binary robust invariant scalable keypoints. International conference on computer vision (ICCV 2011), Barcelona, Spain, IEEE.

Lew, M. S. (2000). "Next-generation web searches for visual content." Computer **33**(11): 46-53.

Li, X. (2003). "Image retrieval based on perceptive weighted color blocks." Pattern Recognition Letters, Elsevier **24**(12): 1935-1941.

Lindeberg, T. (2013). Image matching using generalized scale-space interest points. International Conference on Scale Space and Variational Methods in Computer Vision, Springer.

Lindeberg, T. (2013). "Scale selection properties of generalized scale-space interest point detectors." Journal of Mathematical Imaging and Vision,Springer **46**(2): 177-210.

Lloyd, S. (1982). "Least squares quantization in PCM." Transactions on Information Theory, IEEE **28**(2): 129-137.

Lowe, D. G. (1999). Object recognition from local scale-invariant features. Proceedings of the Seventh International Conference on Computer Vision (ICCV 1999), Kerkyra, Corfu, Greece, IEEE.

Lu, T.-C. and C.-C. Chang (2007). "Color image retrieval technique based on color features and image bitmap." Information processing & management **43**(2): 461-472.

Ma, W.-Y. and B. S. Manjunath (1999). "Netra: A toolbox for navigating large image databases." Multimedia Systems,Springer **7**(3): 184-198.

MacQueen, J. (1967). Some methods for classification and analysis of multivariate observations. Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability, California, Berkeley, University of California Press.

Manjunath, B. S. and W.-Y. Ma (1996). "Texture features for browsing and retrieval of image data." Transactions on Pattern Analysis and Machine Intelligence (PAMI 1996), IEEE **18**(8): 837-842.

Manjunath, B. S., J.-R. Ohm, V. V. Vasudevan and A. Yamada (2001). "Color and texture descriptors." Transactions on circuits and systems for video technology, IEEE **11**(6): 703-715.

Marasco, E. and A. Ross (2015). "A survey on antispoofing schemes for fingerprint recognition systems." Computing Surveys (CSUR), ACM **47**(2): 1-44.

Montoya-Zegarra, J. A., J. Beeck, N. Leite, R. Torres and A. Falcão (2008). Combining global with local texture information for image retrieval applications. Tenth International Symposium on Multimedia (ISM 2008), Berkeley, California, USA, IEEE.

Morel, J.-M. and S. Solimini (2012). Variational methods in image segmentation: with seven image processing experiments, Springer Science & Business Media.

Nallaperumal, K., M. S. Banu and C. C. Christiyana (2007). Content based image indexing and retrieval using color descriptor in wavelet domain. International Conference on Computational Intelligence and Multimedia Applications (ICCIMA 2007), Sivakasi, Tamilnadu, India, IEEE.

Niblack, C. W., R. Barber, W. Equitz, M. D. Flickner, E. H. Glasman, D. Petkovic, P. Yanker, C. Faloutsos and G. Taubin (1993). QBIC project: querying images by content, using color, texture, and shape. Symposium on Electronic Imaging: Science and Technology, IS&T/SPIE.

Nölle, M. (2003). Distribution distance measures applied to 3-d object recognition–a case study. Symposium on Joint Pattern Recognition, Berlin Heidelberg, Springer.

Nurhaida, I., R. Manurung and A. M. Arymurthy (2012). Performance comparison analysis features extraction methods for batik recognition. International Conference on Advanced Computer Science and Information Systems (ICACSIS 2012), Indonesia, IEEE.

Ohanian, P. P. and R. C. Dubes (1992). "Performance evaluation for four classes of textural features." Pattern Recognition, Elsevier **25**(8): 819-833.

Ojala, T., T. Maenpaa, M. Pietikainen, J. Viertola, J. Kyllonen and S. Huovinen (2002). Outex-new framework for empirical evaluation of texture analysis algorithms. Proceedings of the 16th International Conference on Pattern Recognition (ICPR 2002), Quebec, Canada, IEEE.

Ojala, T., M. Pietikäinen and D. Harwood (1996). "A comparative study of texture measures with classification based on featured distributions." Pattern Recognition, Elsevier **29**(1): 51-59.

Ojala, T., M. Pietikainen and T. Maenpaa (2002). "Multiresolution gray-scale and rotation invariant texture classification with local binary patterns." IEEE Transactions on pattern analysis and machine intelligence **24**(7): 971-987.

Ortega, M., Y. Rui, K. Chakrabarti, S. Mehrotra and T. S. Huang (1997). Supporting similarity queries in MARS. Proceedings of the fifth ACM international conference on Multimedia, Seattle, WA, USA, ACM.

Palm, C. (2004). "Color texture classification by integrative co-occurrence matrices." Pattern Recognition, Elsevier **37**(5): 965-976.

Paschos, G., I. Radev and N. Prabakar (2003). "Image content-based retrieval using chromaticity moments." Transactions on Knowledge and Data Engineering, IEEE **15**(5): 1069-1072.

Pass, G., R. Zabih and J. Miller (1997). Comparing images using color coherence vectors. Proceedings of the fourth ACM international conference on Multimedia, Seattle, WA, USA, ACM.

Pentland, A. P. (1984). "Fractal-based description of natural scenes." Transactions on Pattern Analysis and Machine Intelligence (PAMI 1984), IEEE(6): 661-674.

Pydipati, R., T. Burks and W. Lee (2006). "Identification of citrus disease using color texture features and discriminant analysis." Computers and Electronics in Agriculture, Elsevier **52**(1): 49-59.

Rao, M. B., D. B. P. Rao and A. Govardhan (2011). "Content based image retrieval using dominant color, texture and shape." International Journal of Engineering Science and Technology (IJEST) **3**(4).

Reddy, P. G. (2010). Extraction of image features for an effective CBIR system. Recent Advances in Space Technology Services and Climate Change (RSTSCC), 2010, IEEE.

Richards, W. and A. Polit (1974). "Texture matching." Kybernetik **16**(3): 155-162.

Rittscher, J., R. Machiraju and S. T. Wong (2008). Microscopic image analysis for life science applications, Artech House.

Ro, Y. M. and H. K. Kang (2000). "Hierarchical rotational invariant similarity measurement for MPEG-7 homogeneous texture descriptor." Electronics Letters, IEEE **36**(15): 1.

Rojas, R. (2013). Neural networks: a systematic introduction, Springer Science & Business Media.

Rublee, E., V. Rabaud, K. Konolige and G. Bradski (2011). ORB: An efficient alternative to SIFT or SURF. International Conference on Computer Vision (ICCV 2011), Barcelona, Spain, IEEE.

Rubner, Y. and C. Tomasi (1999). Texture-based image retrieval without segmentation. Proceedings of the Seventh International Conference on Computer Vision (ICCV1999) Kerkyra, Corfu, Greece, IEEE.

Selesnick, I. W., R. G. Baraniuk and N. C. Kingsbury (2005). "The dual-tree complex wavelet transform." signal Processing Magazine, IEEE **22**(6): 123-151.

Shearer, S. A. and R. Holmes (1990). "Plant identification using color co-occurrence matrices." Transactions of the ASAE **33**(6): 1237-1244.

Shim, S.-O. and T.-S. Choi (2003). Image indexing by modified color cooccurrence matrix. Proceedings of the International Conference on Acoustics, Speech, and Signal Processing (ICASSP 2003), Hong Kong, IEEE.

Shrivakshan, G. and C. Chandrasekar (2012). "A comparison of various edge detection techniques used in image processing." IJCSI International Journal of Computer Science Issues **9**(5): 272-276.

Shuttleworth, J. K., A. G. Todman, R. N. Naguib, B. M. Newman and M. K. Bennett (2002). Colour texture analysis using co-occurrence matrices for classification of colon cancer images. Canadian Conference on Electrical and Computer Engineering (CCECE 2002). , Winnipeg, Manitoba, Canada, IEEE.

Sim, D.-G., H.-K. Kim and R.-H. Park (2001). "Fast texture description and retrieval of DCT-based compressed images." Electronics Letters, IEEE **37**(1): 1.

Singh, S. and M. Sharma (2001). Texture analysis experiments with meastex and vistex benchmarks. International Conference on Advances in Pattern Recognition (ICAPR 2001), Rio de Janeiro, Brazil, Springer.

Sklansky, J. (1978). "Image segmentation and feature extraction." IEEE Transactions on Systems, Man, and Cybernetics **8**(4): 237-247.

Smith, J. R. and S.-F. Chang (1996). Automated binary texture feature sets for image retrieval. Proceedings of the International Conference on Acoustics, Speech, and Signal Processing (ICASSP 1996), Munich, Germany, IEEE.

Smith, J. R. and S.-f. Chang (1997). "Querying by color regions using the VisualSEEk content-based visual query system." Intelligent multimedia information retrieval **7**(3): 23-41.

Stanchev, P., D. Green Jr and B. Dimitrov (2003). "High level color similarity retrieval." International Journal ITA **10**(3).

Stehling, R. O., M. A. Nascimento and A. X. Falcão (2001). An adaptive and efficient clustering-based approach for content-based image retrieval in image databases. International Symposium on Database Engineering and Applications (IDEAS 2001), Grenoble, France, IEEE.

Stehling, R. O., M. A. Nascimento and A. X. Falcão (2002). A compact and efficient image retrieval approach based on border/interior pixel classification. Proceedings of the Eleventh International Conference on Information and Knowledge Management (CIKM 2002), McLean, Virginia, USA, ACM.

Stehling, R. O., M. A. Nascimento and A. X. Falcão (2003). "Cell histograms versus color histograms for image representation and retrieval." Knowledge and Information Systems,Springer **5**(3): 315-336.

Stricker, M. A. and M. Orengo (1995). Similarity of color images. Symposium on Electronic Imaging: Science & Technology, IS&T/SPIE.

Sun, J., X. Zhang, J. Cui and L. Zhou (2006). "Image retrieval based on color distribution entropy." Pattern Recognition Letters, Elsevier **27**(10): 1122-1126.

Swain, M. J. and D. H. Ballard (1991). "Color indexing." International Journal of Computer Vision,Springer **7**(1): 11-32.

Talukder, K. H. and K. Harada (2010). "Haar wavelet based approach for image compression and quality assessment of compressed image." arXiv preprint arXiv:1010.4084.

Tamura, H., S. Mori and T. Yamawaki (1978). "Textural features corresponding to visual perception." Transactions on Systems, Man, and Cybernetics, IEEE **8**(6): 460-473.

Tan, T. and J. Kittler (1994). "Colour texture analysis using colour histogram." Proceedings-Vision, Image and Signal Processing, IEEE **141**(6): 403-412.

Tao, B. and B. W. Dickinson (2000). "Texture recognition and image retrieval using gradient indexing." Journal of Visual Communication and Image Representation, Elsevier **11**(3): 327-342.

The Center for Machine Vision and Signal Analysis (CMVS). (21 November 2014). "Browse Image Database." Retrieved 29/7/2016, from http://www.outex.oulu.fi/index.php?page=browse.

The Center for Machine Vision and Signal Analysis (CMVS). (21 November 2014). "Texture classification test suites." Retrieved 29/7/2016, from http://www.outex.oulu.fi/index.php?page=classification.

Tuceryan, M. and A. K. Jain (1993). "Texture analysis." Handbook of pattern recognition and computer vision **2**: 207-248.

Unser, M. (1986). "Local linear transforms for texture measurements." Signal Processing, Elsevier **11**(1): 61-79.

Unser, M. (1986). "Sum and difference histograms for texture classification." Transactions on Pattern Analysis and Machine Intelligence (PAMI 1986), IEEE(1): 118-125.

Vadivel, A., S. Sural and A. K. Majumdar (2005). Human color perception in the HSV space and its application in histogram generation for image retrieval. Symposium on Electronic Imaging, International Society for Optics and Photonics.

Vadivel, A., S. Sural and A. K. Majumdar (2007). "An integrated color and intensity co-occurrence matrix." Pattern recognition Letters, Elsevier **28**(8): 974-983.

Valkealahti, K. and E. Oja (1998). "Reduced multidimensional co-occurrence histograms in texture classification." Transactions on Pattern Analysis and Machine Intelligence (PAMI 1998), IEEE **20**(1): 90-94.

Van Der Zwan, R., A. Kukulska-Hulme, T. Dipaolo, V. Evers and S. Clarke (1999). "The Informedia digital video library system at the Open University." Proceedings of the CIR-99: The Challenge of Image Retrieval.

Veltkamp, R., H. Burkhardt and H.-P. Kriegel (2013). State-of-the-art in content-based image and video retrieval, Springer Science & Business Media.

Wactlar, H. D., T. Kanade, M. A. Smith and S. M. Stevens (1996). "Intelligent access to digital video: Informedia project." Computer **29**(5): 46-52.

Wandell, B. A. (1996). "Book Rvw: Foundations of Vision. By Brian A. Wandell." Journal of Electronic Imaging (JEI), SPIE **5**(1): 107-107.

Wang, J. Z., J. Li, G. Wiederhold and O. Firschein (1998). Classifying objectionable websites based on image content. International Workshop on Interactive Distributed Multimedia Systems and Telecommunication Services (IDMS 1998), Oslo, Norway, Springer.

Wang, J. Z., G. Wiederhold, O. Firschein and S. X. Wei (1998). "Content-based image indexing and searching using Daubechies' wavelets." International Journal on Digital Libraries **1**(4): 311-328.

Wang, Q., D. D. Feng and Z. Chi (2004). B-spline over-complete wavelet based fractal signature analysis for texture image retrieval. Proceedings of the International Symposium on Intelligent Multimedia, Video and Speech Processing, Hong Kong, IEEE.

Wei, G., D. Li and I. K. Sethi (1998). Web-WISE: Compressed image retrieval over the Web. Multimedia Information Analysis and Retrieval, Springer**:** 33-46.

Williams, A. and P. Yoon (2007). "Content-based image retrieval using joint correlograms." Multimedia Tools and Applications,Springer **34**(2): 239-248.

Wong, K. M., L. M. Po and K. W. Cheung (2007). A compact and efficient color descriptor for image retrieval. Proceedings of the International Conference on Multimedia and Expo (ICME 2007), Beijing, China, IEEE.

Wu, P., B. Manjunath, S. Newsam and H. Shin (2000). "A texture descriptor for browsing and similarity retrieval." Signal Processing: Image Communication, Elsevier **16**(1): 33-43.

Wu, Y. and Y. Wu (2009). Shape-based image retrieval using combining global and local shape features. 2nd International Congress on Image and Signal Processing (CISP 2009) IEEE.

Yaglom, A. M. (2004). An introduction to the theory of stationary random functions, Courier Corporation.

Yang, D., J. H. Garrett Jr, D. S. Shaw and L. A. Rendell (1994). "An intelligent symbol usage assistant for CAD systems." Expert: Intelligent Systems and Their Applications, IEEE **9**(3): 32-41.

Yang, M., K. Kpalma and J. Ronsin (2008). "A survey of shape feature extraction techniques." Pattern Recognition, Elsevier: 43-90.

Yang, M., K. Kpalma and J. Ronsin (2008). "A survey of shape feature extraction techniques." Pattern recognition: 43-90.

Yang, N.-C., W.-H. Chang, C.-M. Kuo and T.-H. Li (2008). "A fast MPEG-7 dominant color extraction with new similarity measure for image retrieval." Journal of Visual Communication and Image Representation, Elsevier **19**(2): 92-105.

Yang, X. and J. Liu (2002). "Maximum entropy random fields for texture analysis." Pattern recognition Letters, Elsevier **23**(1): 93-101.

Zhang, D. and G. Lu (2004). "Review of shape representation and description techniques." Pattern Recognition, Elsevier **37**(1): 1-19.

Zhou, J., L. Xin and D. Zhang (2003). "Scale-orientation histogram for texture image retrieval." Pattern Recognition, Elsevier **36**(4): 1061-1063.

Zhu, S.-C., C.-E. Guo, Y. Wang and Z. Xu (2005). "What are textons?" International Journal of Computer Vision,Springer **62**(1-2): 121-143.

Zucker, S. and K. Kant (1981). Multiple-level representations for texture discrimination. Conference on Pattern Recognition and Image Processing, Dallas, TX.