ANALYSIS OF TEXTURAL IMAGE FEATURES FOR CONTENT BASED RETRIEVAL

by

ERAY KULAK

Submitted to the Graduate School of Engineering and Natural Sciences
in partial fulfillment of
the requirements for the degree of
Master of Science

Sabancı University
October 2002

ANALYSIS OF TEXTURAL IMAGE FEATURES
FOR CONTENT BASSED RETRIEVAL

APPROVED BY:

Prof. Dr. Aytül Erçil …………………………….
(Thesis Supervisor)

Dr. Yücel Saygın …………………………….

Dr. Yaşar Gürbüz …………………………….

DATE OF APPROVAL: …………………………….

To the memory of Gazi Mustafa Kemal ATATÜRK,

ACKNOWLEDGMENTS

I would like to thank my thesis supervisor, Prof. Dr. Aytül Erçil, for her guidance throughout the Master of Science study, for the valuable suggestions, and for sharing her excellent archive.

Special thanks to my hard working colleagues, Özgür Kulak and Çiğdem Altay.

I wish to remember Alper Atalay whose studies helped us a lot.

I would also like to thank Dr. Yaşar Gürbüz and Dr. Yücel Saygın for participating in my thesis committee, and motivating me on the study.

I am forever grateful to my parents, Mualla and Bekir Kulak, for their unconditional love and support.

ANALYSIS OF TEXTURAL IMAGE FEATURES
FOR CONTENT BASED RETRIEVAL

ABSTRACT

Digital archaelogy and virtual reality with archaeological artefacts have been quite hot research topics in the last years[55,56]. This thesis is a preperation study to build the background knowledge required for the research projects, which aim to computerize the reconstruction of the archaelogical data like pots, marbles or mosaic pieces by shape and textural features.

Digitalization of the cultural heritage may shorten the reconstruction time which takes tens of years currently[61]; it will improve the reconstruction robustness by incorporating with the literally available machine vision algorithms and experiences from remote experts working on a no-cost virtual object together. Digitalization can also ease the exhibition of the results for regular people, by multiuser media applications like internet based virtual museums or virtual tours. And finally, it will make possible to archive values with their original texture and shapes for long years far away from the physical risks that the artefacts currently face.

On the literature[1,2,3,5,8,11,14,15,16], texture analysis techniques have been throughly studied and implemented for the purpose of defect analysis purposes by image processing and machine vision scientists. In the last years, these algorithms have been started to be used for similarity analysis of content based image retrieval[1,4,10]. For retrieval systems, the concurrent problems seem to be building efficient and fast systems, therefore, robust image features haven't been focused enough yet. This document is the first performance review of the texture algorithms developed for retrieval and defect analysis together. The results and experiences gained during the thesis study will be used to support the studies aiming to solve the 2D puzzle problem using textural continuity methods on archaelogical artifects, Appendix A for more detail.

The first chapter is devoted to learn how the medicine and psychology try to explain the solutions of similiarity and continuity analysis, which our biological model, the human vision, accomplishes daily.

In the second chapter, content based image retrieval systems, their performance criterias, similiarity distance metrics and the systems available have been summarized.

For the thesis work, a rich texture database has been built, including over 1000 images in total. For the ease of the users, a GUI and a platform that is used for content based retrieval has been designed; The first version of a content based search engine has been coded which takes the source of the internet pages, parses the metatags of images and downloads the files in a loop controlled by our texture algorithms. The preprocessing algorithms and the pattern analysis algorithms required for the robustness of the textural feature processing have been implemented. In the last section, the most important textural feature extraction methods have been studied in detail with the performance results of the codes written in Matlab and run on different databases developed.

ANALYSIS OF TEXTURAL IMAGE FEATURES FOR CONTENT BASED RETRIEVAL

ÖZET

Sayısal arkeoloji ve sanal gerçeklik uygulamaları ile arkeoloji verilerinin birleştirilmesi, son yıllarda bilimsel ilgi çeken konulardır[55,56]. Tez çalışmamız, çömlek parçaları, mermer röliyef veya mozaikler şeklindeki arkeolojik bulguların, şekil ve doku bilgileri kullanılarak bilgisayar desteği ile ilişkilendirilmesini hedefleyen projelere ön araştırma niteliğindedir.

Kültürel mirasın sayısallaştırılarak kullanılması, yıllar süren[61] restorasyon çalışmalarını kısaltacak, geri dönülebilir alternatifler üzerinde daha çok uzman fikrin paylaşılmasını ve yapay görme sistemleriyle de sonuçların gürbüzlüğünü sağlayacaktır. Ayrıca, sonlanan çalışmalardan elde edilen veriler sanal müze, sanal tur gibi çoklu medya uygulamaları ile elektronik ortamlarda kitlelere ulaştırılabilecek, ve objeler yıllar boyunca fiziksel risklerden uzak arşivlenebilecektir.

Doku analizi günümüze kadar hata analizi amacıyla yapay görme ve imge işleme bilimlerinde sıkça araştırılmış ve yaygın olarak uygulanmıştır[1,2,3,5,8,11,14,15,16]. Son yıllarda ise benzerlik analizi alanına kayarak içerik tabanlı sayısal imge arama yöntemlerinde kullanılmaya başlanmıştır[1,4,10]. Bu sistemlerin güncel problemleri verimli yapı ve hız özellikleri olduğundan imge metrikleri üzerinde henüz fazla yoğunlaşılmamıştır. Tez çalışmamız hata ve benzerlik analizlerindeki doku algoritmalarının sayısal imge arama platformunda toplu haldeki ilk karşılaştırma dökümanıdır. Benzerlik analizinden çıkan veriler ve tez çalışmasında elde edilen deneyimler 2 boyutlu bulmaca çözme, bir başka ismiyle devamlılık analizi yöntemlerini geliştirmek üzere ilk kez uygulancak olan arkeolojik verilerin doku metrikleri ile ilişkilendirilmesi çalışmalarında kullanılacaktır.

Tezin ilk bölümünde problemlerimize model teşkil edecek insan görme sistemi incelenmiş ve biyolojik olarak hata, benzerlik ve devamlılık analizlerinin çözümlerinin tıp ve psikoloji bilimlerince nasıl açıklandığı araştırılmıştır.

İkinci bölümde, sayısal içerik tabanlı 2 boyutlu durağan imge arama sistemleri, performans kriterleri ve yakınlık metrikleri incelenmiş, literatürdeki çalışmalar özetlenmiştir.

Tez çalışması için, literatürde şimdiye kadar kullanılan en geniş doku örnek arşivi oluşturulmuştur. Doku analizi sonuçlarının görsel olarak izlenebileceği bir arayüz tasarlanmış, içerik tabanlı imge arama yapılabilecek bir platform geliştirilmiş, internet sayfaları kaynak kodlarından karakter seti resim bileşeni ayrıştırması yaparak resim dosyalarını bulup sabit diske indirecek içerik tabanlı arama motoru kod çalışmasının ilk yapısı uygulanmıştır; Ayrıca, doku analizi yöntemleri öncesi gerekli olan imge işleme ve doku metriklerinin gürbüz işlenmesi için örüntü analizi yöntemleri gerçekleştirilmiştir. Son bölümde ise geliştirilen en yaygın doku analizi ve bu yöntemlerin tüm metrikleri açıklanmış, yaratılan farklı boyutlardaki doku örnek arşivlerinde, yazılan kodların performans sonuçları özetlenmiştir.

TABLE OF CONTENTS

TABLE OF CONTENTS (continued)

LIST OF TABLES

LIST OF FIGURES

LIST OF FIGURES (continued)

LIST OF FIGURES (continued)

CHAPTER 1

INTRODUCTION

1.1. Problem Statement

The aim of this thesis is to study content-based static image retrieval systems and implement literally available, successful textural feature extraction algorithms. The knowledge obtained will guide a further research for the texture-based solution of the 2D-puzzle problems to computerise the reconstruction of archaeological artefacts.

Let’s clarify this compact problem statement a little:

First of all, what does the term “texture” mean? There are many definitions available in the literature[37], but no unique one is agreed upon. For simplicity, we include all the information within the object boundary under the texture title. The colour, edginess, smoothness, coarseness, periodicity... all the discriminative visual feeling we get from the surface of the object constructs the textural feature set for our image retrieval system.

Why do we need content-based systems? Up to recent years, the most data available in the computer world were text and number based. The new MPEG standards, rich Internet environment, image databases of the museums, archives of medicine increase multimedia information day by day, which make text-based descriptive indexing impossible for the advanced searches and associations. Therefore, content-based image retrieval systems were initiated in the 1980s[56,62]. Our concern in this field is to build a benchmark platform to compare the performance of various texture algorithms.

This thesis will guide a further research on “puzzle problems” of the vision literature, as explained in detail at the Appendix A. The artefacts found in archaeological sites create such real world puzzles. Traditionally, experts analyse the pieces mainly by their bare eyes and try to find associations to extract neighbourhoods.

The reconstruction task sometimes faces with tremendous amount of unstructured data, which may ask years[61] to cope with. More important, by such a manual search, the alternatives during the reconstruction are usually ignored to reach one possible shortest solution. All the data is non-digital, and for this reason, the information sharing among remote professionals, visualisation before the final reconstruction and utilising from the features other than shape and texture are usually skipped. Computerising the solution of the puzzle problem will definitely benefit the archaeology world to gain effort and efficiency.

1.2. Outline of the Thesis

For the content retrieval and jigsaw puzzles, one possible solution of designing a powerful machine vision system is to mimic the nature as a model: The human eye decomposes the objects seen into piecewise feature windows including edge, texture, colour, shading, direction, etc. Then the brain composes the global picture again, associates with the memories and attaches a meaning to the object, exactly like solving a puzzle by content features. What we try to do is to replicate this biological solution in our computer systems with the concern to improve textural type of feature. In short, analysis of eye and brain pair will help us to show what we should care and what kind of complexity we face. Therefore the first chapter is devoted for the biological vision.

The second chapter introduces us the available computerised content-based systems. After a complete summary, we will focus on how they process the textural information.

In the last chapters 5 and 6, the textural algorithms and their retrieval performances are evaluated. The main target is to find an appropriate algorithm with enough discrimination power for the first level content based image retrieval, with a general 2D puzzle problem in mind.

Finally, the last section concludes the study and summarises shortly what other issues are needed in future to transfer the similarity analysis into a continuity analysis.

CHAPTER 2

HUMAN VISION

This chapter explores the human vision system, the pair of eye and brain. We'll first prove[6] the partnership of the brain to the eye on the vision process and describe how the visual information signals reach to the surface of the brain. Then, some rules governing the perception will be exampled to show that the vision process doesn't end with the image acquisition. Next, we let the reader face with the complexity of the two piece puzzle problem when it is attempted to be solved by the computers. Finally, we will focus on what happens to the extracted feature windows in the brain, and how they are reassembled.

2.1. The Partnership of the Brain and Eye

The human vision is usually thought[6,7,17,18,24,25] with its processing unit, the brain, unified.

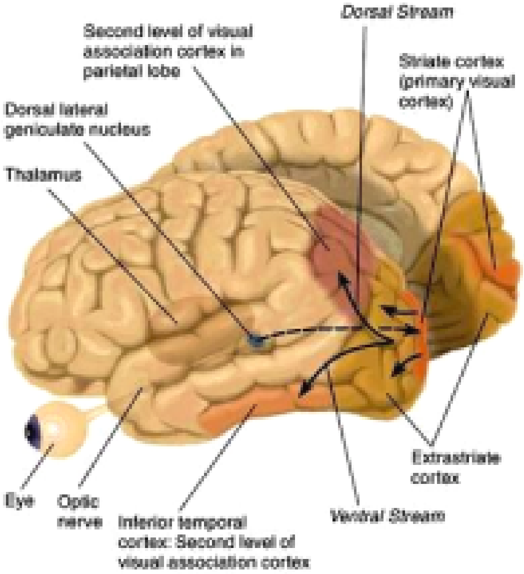

Dorsal Stream
Second level of visual association cortex in parietal lobe
Striate cortex (primary visual cortex)
Dorsal lateral geniculate nucleus
Thalamus
Extrastriate cortex
Eye
Optic nerve
Inferior temporal cortex: Second level of visual association cortex
Ventral Stream

Figure 2.1: The human visual system

The eyes take the pictures like a camera, and different brain regions compose a meaning, and identify the object or the scene. Both systems depend on each other. A detailed vision process will be explained soon, but we'll first prove this collaboration between the eye and brain by a typical experiment[17] from the cognitive sciences:

A californian woman, N.G., was placed in front of a screen that had a small black dot in the centre. She was asked to fix her eyes on this dot, thus ensuring that images entering her eyes from the side were sent to one hemisphere only. The experimenter then briefly flashed a picture of a cup to the right of the dot. The image stayed on the screen for a very short while -about a twentieth of a second. This is just long enough for an image to be registered, but not long enough for a person to shift her eyes to bring it into focus and thus send it to both hemispheres. So the image on the screen went to N.G.'s left hemisphere only and there it stopped, because the normal route to her other hemisphere, the corpus callosum, was cut.

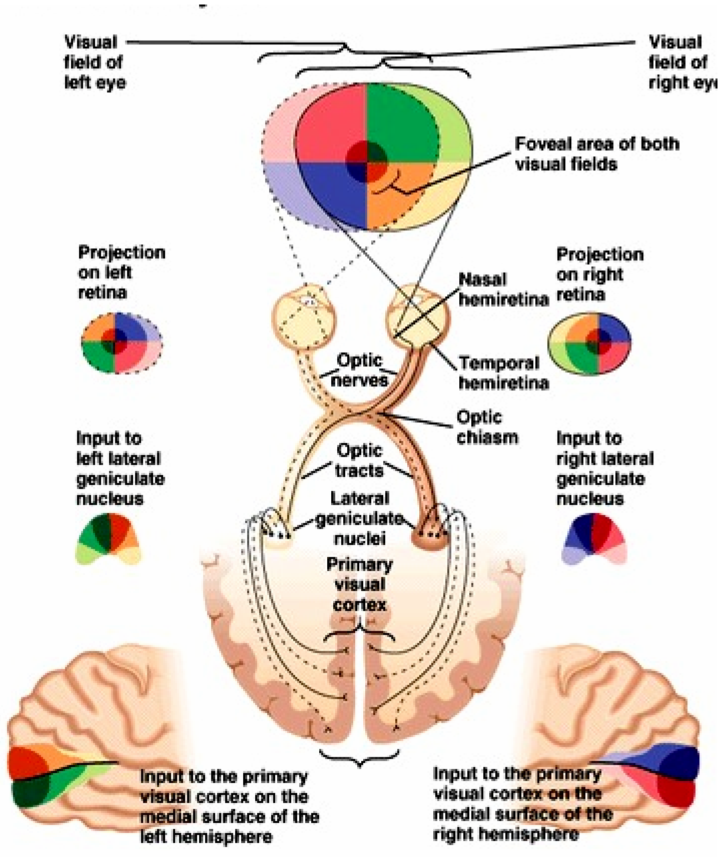

Visual field of left eye
Visual field of right ey
Foveal area of both visual fields
Projection on left retina
Nasal hemiretina
Projection on right retina
Optic nerves
Temporal hemiretina
Optic chiasm
Input to left lateral geniculate nucleus
Optic tracts
Input to right lateral geniculate nucleus
Lateral geniculate nuclei
Primary visual cortex
Input to the primary visual cortex on the medial surface of the left hemisphere
Input to the primary visual cortex on the medial surface of the right hemisphere

Figure 2.2: The retina-geniculate-striate system

Asked what she saw, N.G. replied, quite normally: 'A cup'. Next, a picture of a spoon was flashed on the left side of the screen so that it entered N.G.'s right hemisphere. This time, when she was asked what she had seen, she said: 'Nothing'. The experimenter then asked N.G. to reach under the screen with her left hand and select, by touch only, from among a group of concealed items the one that was the same as the one she had just seen. She felt around, pausing briefly at a cup, a knife, a pen, a comb, then settled firmly on a spoon. While her hand continued to hold it behind the screen, the experimenter asked her what she was holding: 'A pencil' she said. These responses, though inexplicable at first sight, actually gave the researchers a uniquely clear picture of what was going on in N.G.'s brain:

When the cup image was sent to the left hemisphere she saw it and named it in the usual way. When the spoon image was fed to her right hemisphere, however, she could not tell the experimenter about it because her right hemisphere was unable to speak. The words that came out of her mouth, 'I see nothing', were uttered by the left hemisphere, the only one that could reply. And, because the left hemisphere is isolated from the right, it was speaking quite truthfully, it had no knowledge of the spoon picture because, without the usual passage of information across the corpus callosum, the image never reached it. That does not mean that the spoon image did not go in, however. When the subject was asked to use her left hand to select an item it was the knowledge in the right hemisphere which is wired to the left hand that was brought to bear on the task. Thus she selected the spoon. But when the experimenter asked her to name it she was up against the same problem she encountered when asked to name the image of the spoon, the right hemisphere could not tell of it. Instead the left hemisphere kicked in and did the logical thing. Because it was unaware of the spoon image, it had no way of knowing that the left hand had selected a spoon rather than any other item. It could not see the spoon because the hand that grasped it was under the screen. And it couldn't feel it because the sensory stimuli from the left hand were going, as normal, to the right hemisphere, where it stayed in the absence of the carrus callosum. The left hemisphere knew that something was in the left hand, but it had to identify it by guesswork or deduction. As it happens the left hemisphere is pretty good at deduction, and it calculated that of all the objects that might be tucked away behind the screen a pencil seemed like a good bet. So 'pencil' it said.

The spoon and cup exercises demonstrated that the vision process doesn't end with the image acquisition step, brain sensory areas add value to what seen.

Sensory processing employs the vast majority of the brain cortex, only the frontal lobes are dedicated to non-sensual tasks. The partition is almost the same for everybody but excessive use of a single sense can cause the relevant cortical area to expand, just like a muscle when it is exercised. The vision area is at the backside of the brain.[17]

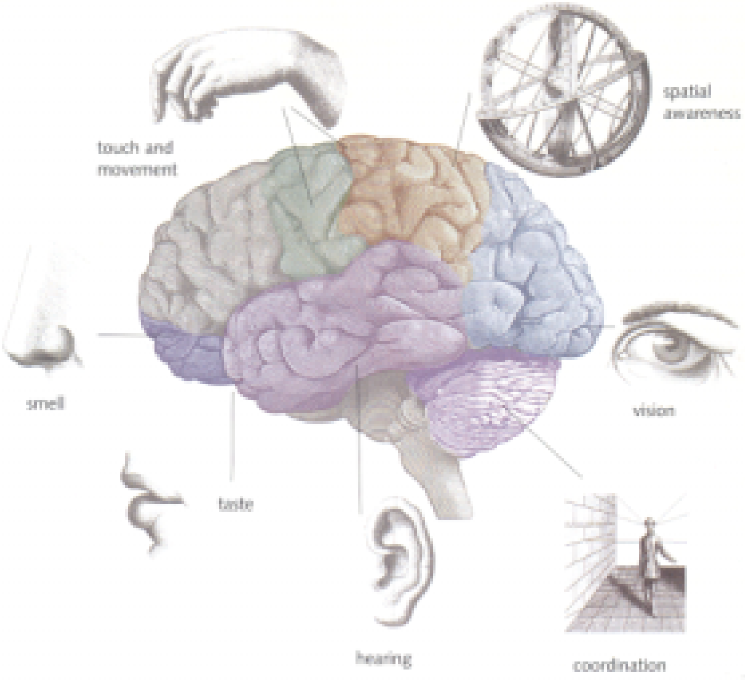


Figure 2.3: The senses of the brain and cortex task areas[17]

2.2. The Vision Process from Light to the Brain

The vision process starts[25,18,17,7,23] when the light from a visual stimulus is inverted as it passes through the lens. The light then hits the back of the eye, where light-sensitive cells turn it into a message of electrical pulses.

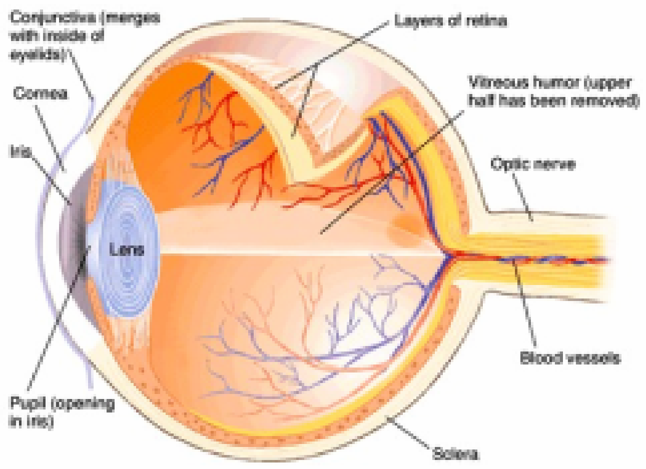

Conjunctiva (merges with inside of eyelids)
Cornea
Iris
Lens
Pupil (opening in iris)
Layers of retina
Vitreous humor (upper half has been removed)
Optic nerve
Blood vessels
Sclera

Figure 2.4: The structure of the eye

To look closely at something is to turn one's eyes so that the image falls near the centre of the retina, a specialised area smaller than the head of a pin that is named as fovea. Only in this tiny region, the receptor cells concentrated with sufficient density provide detailed vision. As a result not more than a thousandth of the entire visual field can be seen in "hard focus" at a given moment.

Yet the human eye is capable of discerning in considerable detail a scene as complex and swiftly changing as the one confronting a person driving an automobile in traffic. This formidable visual task can be accomplished only because the eyes are able to flick rapidly about the scene, with the two foveae receiving detailed images first from one part of the scene and then from another. Therefore, most of the times our eyes are jumping from point to point, each fixation takes only a few milliseconds. This rapid jump is the most common major eye movement and named as the saccade.

The duration of the fixations depends on the character of the scene and what the viewer is doing. The flick may be so rapid that the eye's angular velocity may reach more than 500 degrees per second. This velocity is not under conscious control; an effort to slow it will only break the saccade into a series of shorter movements.

If at the end of the saccade the fovea is not "on target," the eye adjusts by making one or more small corrective jumps. The path the eyes follow between two fixation points may be straight, curved or even hooked, but once the eye is launched on a saccade it cannot change its target. It is as if the points in the visual field were recorded as a set of coordinates in the brain. The difference between the coordinates of fixation point at one instant and the next fixation point constitutes an error signal to the eye-movement control centres, and the resulting movement of the eye is directed in a manner analogous to what an engineer would call a simple position servo mechanism.

At each saccade, messages of electrical pulses from light-sensitive cells are carried along the optic nerve from each eye and cross over at the optic chiasma, a major anatomical landmark. The optic track then carries the information to the lateral geniculate body, part of the thalamus.

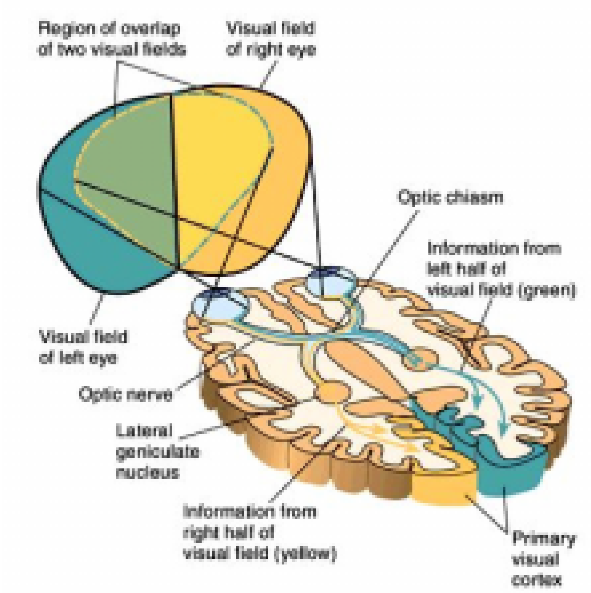

Region of overlap of two visual fields
Visual field of right eye
Optic chiasm
Information from left half of visual field (green)
Visual field of left eye
Optic nerve
Lateral geniculate nucleus
Information from right half of visual field (yellow)
Primary visual cortex

Figure 2.5: The primary visual pathway

This shunts the signal on to V1 at the back of the brain, the vision area. The visual cortex is also split into many areas, each processing an aspect of sight, such as colour, shape, size and so on.

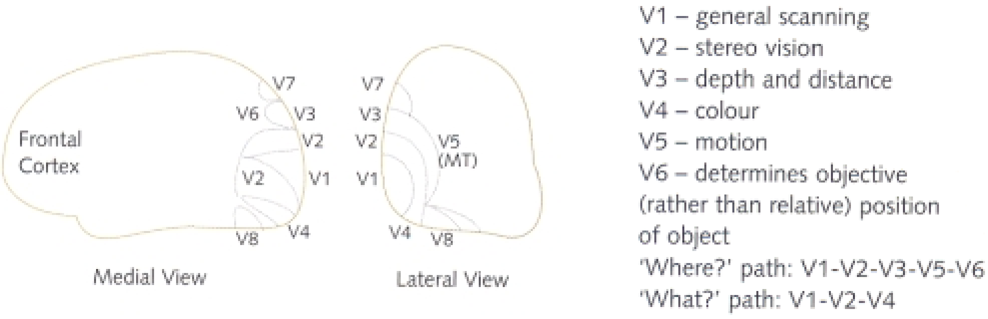


Figure 2.6: The map of the cortex[17]

V1 mirrors the world outside in which each point in the external visual field matches a corresponding point on the V1 cortex. When a person stare at a simple pattern like a grating the image is reflected by a matching pattern of neural activity on the surface of the brain.

2.3. Perception Rules of the Brain

In the last section, we tracked the light up to the brain surface. It is medically hard to work beyond to the cortex[7]. Instead, the physiologists and psychologists proposed[19,17,18,6,7] some vision tests, hypothesising the brain as a black box problem which is solvable by its input and output functions. A few of them are exampled below to show how complex content-based rules are employed in the brain:

The law of pragnanz (Gestalt Laws[19,7,6]): This law says that we are innately driven to experience things in as good a gestalt, form, as possible. “Good” can mean many things here, such a regular, orderly, simplicity, symmetry...

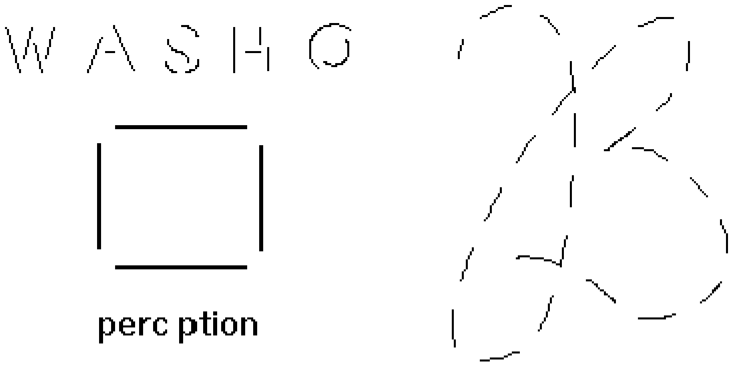


Figure 2.7: The pragnanz example

On the figure above, we can still read 'WASHO', see the square and read 'perception' despite the missing information. At the right, we tend to complete the figure, make it the way it “should” be, and manage to see this as a "B"...

The law of similarity (Gestalt Laws): The dots are seen as horizontal lines because those forming horizontal lines are more similar than those forming vertical lines.

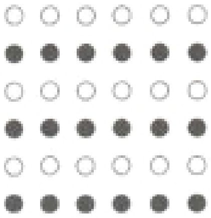


Figure 2.8: The similar lines

Occluding surfaces[18]*:* The perceptual system decides what is hiding and what is hidden easily. Although the features don’t change on the figure below, our perception about the content differs much.

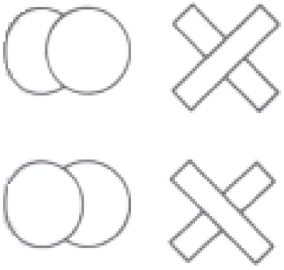


Figure 2.9: The perception of the occluding surfaces

A Bare Bear[6]*:* In this example, the evidence is enough to elicit a ‘conceptual’ hypothesis of a bear, but perhaps not sufficient for actually seeing it perceptually.

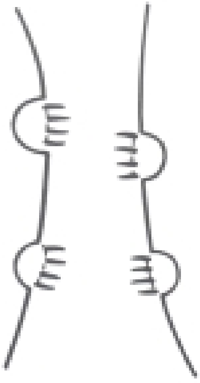


Figure 2.10: The conceptual image

The law of enclosure (Gestalt Laws): On the test below, it is much easier to see the vase than the two faces, because enclosed regions tend to be seen as ‘figure’ rather than ‘ground’, an example of natural segmentation.

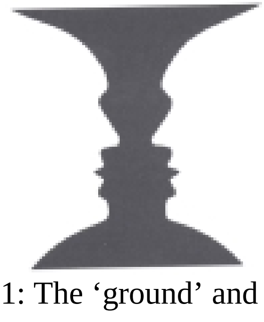


Figure 2.11: The ‘ground’ and ‘figure’

The law of proximity (Gestalt Laws): The dots are seen as arranged in horizontal or vertical lines rather than oblique lines, because they are further apart in the oblique lines.

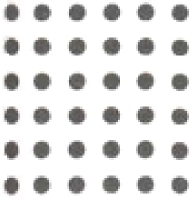


Figure 2.12: The proximity example

The law of good continuity (Gestalt Laws): The small dots below are seen to form a wavy line superimposed on a profile of battlements rather than as the succession of shapes shown at the bottom of the figure. The rule is, we perceive the organisation that interrupts the fewest lines.

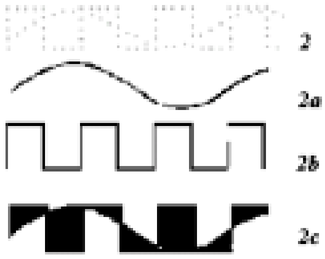


Figure 2.13: The continuity example

Memory associations[6,17]: A patient unable to perceive shape is unable to copy drawings of an apple or an open book, but was much better at drawing these objects from memory, as seen on the figure below. The object memory of the brain helps the vision process much by supported guesses and similarity measures.

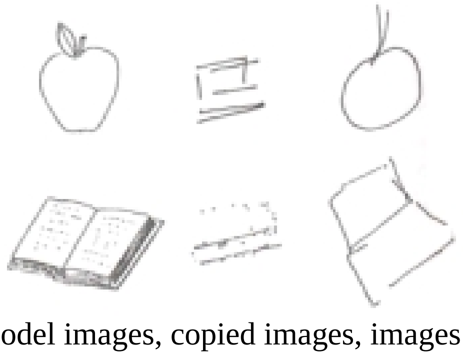


Figure 2.14: The model images, copied images, images drawn from memory

These rules can be extended more, but we understand in short is that content-based image retrieval is not just a feature extraction flow for the human vision system. There are many other rules[6,7,17,18,19,24,25] and many associations utilised by the specific regions of the brain, which are sometimes called as semantic features in content retrieval, at a higher level than the acquisition of the image features by the eyes.

2.4. Two Pieces 2D Puzzle Problem

If we attempt to solve a 2 pieces puzzle by computers, we need the following conditions:

Continuity of colours: Colour continuity is an important factor on the solution of puzzle problems. If asked which combination below is the right solution of the puzzle, the second case seems to be more appropriate for most observers. Usually, the colours are not so homogenous, so right colour spaces, noise filters, shadow eliminators should be used to work with real images.

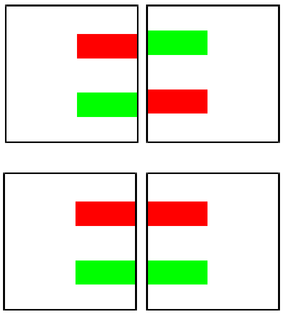


Figure 2.15: The colour continuity

Continuity of edges: The eyes want to see continuous lines, like a derivative operator. The problem here is, usually the broken real object pieces have occluded parts at the boundary zone.

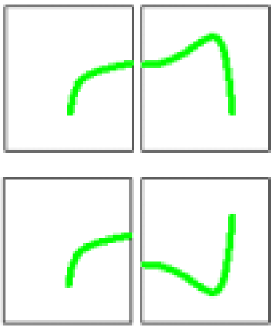


Figure 2.16: The continuity of edges

Continuity of textural information: The two stripes below have a the same colour, but one is horizontal textured and the other is vertical. It is clear that the human vision also tracks the continuity of the textural information like the colour. Therefore, characterising a texture with numerical metrics is needed and as in the edge case, the changes at the boundary affect our success to label the pieces as continuous.

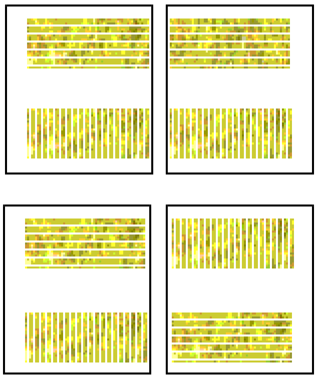


Figure 2.17: The continuity of textures

Object memory: Just to see is not enough, we conceptually add some meaning to the objects; the biological similarity analysis is not 1 or 0 function but a most probable searcher. Below you see two pieces; it cannot be claimed to contain any cues on how these pieces should be associated. In this case, the only decision taker is our memory, which favours the first combination by remembering past experiences. It is probably the most important and difficult factor, if we would like to computerise biological vision systems.

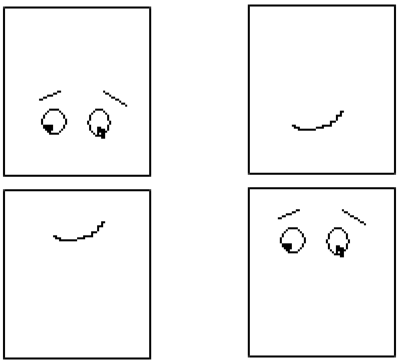


Figure 2.18: The object memory

Boundary matching and generalisation of the problem: Up to now, we decreased the complexity of the puzzle, by regularising the outer boundaries. Let’s use irregular boundaries. Boundary is another dimension for our feature space; the pieces of various sized, occluded and missing parts together increase the complexity exponentially.

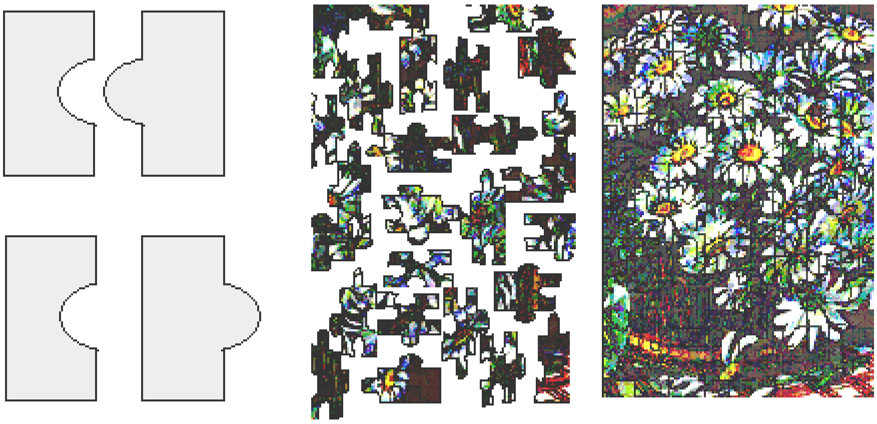


Figure 2.19: The boundary features

The puzzle problem is an exhaustive search problem with dozens of feature windows for each single piece. The problem reveals itself when we remember that the eye supplies 500MB/sec compact feature information[7] for the brain to cope with such ambiguity.

2.5. Ganglion Cells and Neurons

To get an idea how the low-level features are utilised in the vision process, we'll now study the ganglion cells of the retina and neurones of the cortex[7,18,17]:

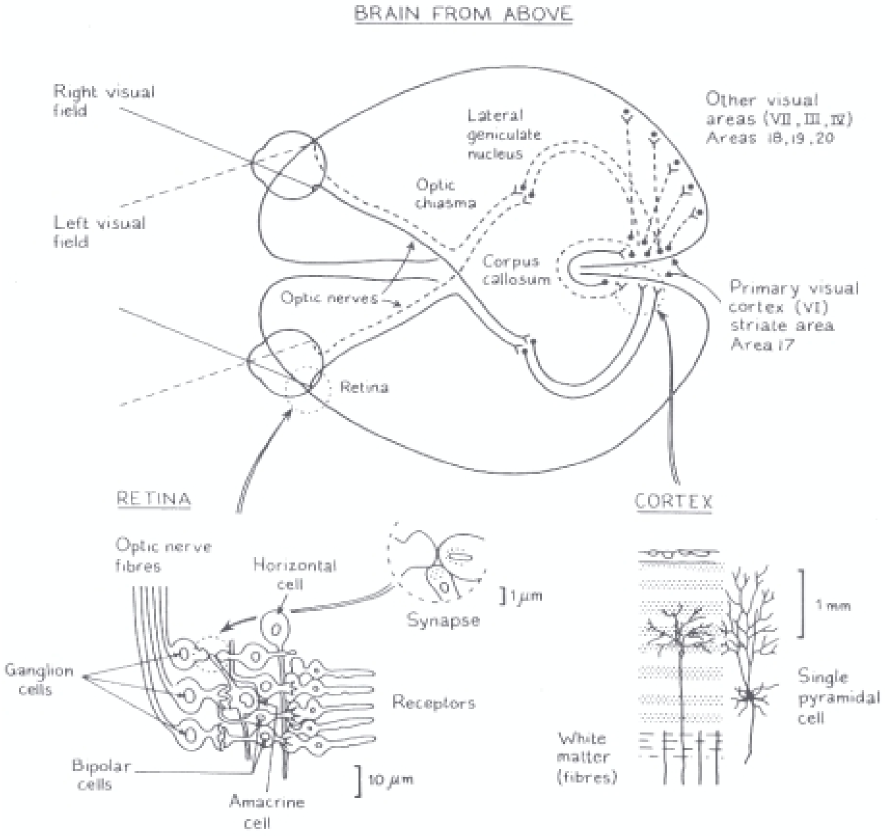


Figure 2.20: The ganglion cells and neurones of the cortex[7]

In the figure above, a portion of the retina is magnified to show the retinal ganglion cells, of which there are 10^6 in each eye. The ganglion cells pick up their messages from the receptors in the retina, through the intermediate cells, called bipolars. The complex synaptic connections between the various cell types in the retina do the computations that determine what property of the light, shade, and colour in the image excites a ganglion cell. The image information pushed to the axons of the ganglion cells is then carried to the brain with one more relay at the lateral geniculate nucleus. In the human retina there are about a million ganglion cells and consequently about a million axons reach to the visual cortex on each side of the brain. To the right of the figure 2.20, a portion of visual cortex is magnified. In the primary visual cortex, there are at least 100 pyramidal cells for each input fibre, density of cells exceeds $10^5/mm^2$. The axons of these cells carry information to the other visual areas of the brain, further in order to employ the perception rules[7].

We may mimic this model on our computerised content-based systems, by first extracting local features like edges, colour, shading, texture and utilising from them through higher level of associations. For the degree of locality, what we know is, the eye is capable of 2-3 degree angle of view for each saccade, during the scanning process[25]. The two-dimensional view angle without distortion is about 30-40 degree. So the proportion should give us approximately, how small our feature extraction area should be. Let us see now, how the ganglion cells extract local features from these small image windows:

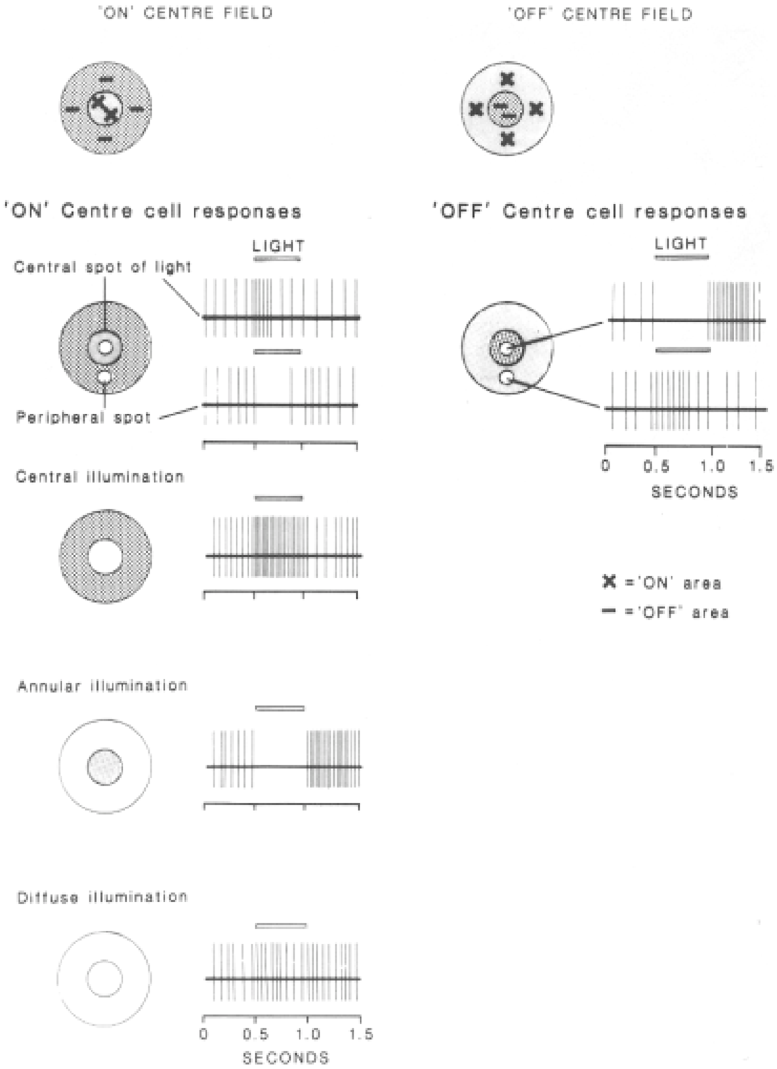


Figure 2.21: Responses from two typical retinal ganglion cells[7]

Figure 2.21 shows the responses of two typical retinal ganglion cells, the type of nerve cell that takes part in transmitting the retinal image up the optic nerve to the brain. One cell deals with one small part of the retina, so in order to excite it the light must be placed in exactly the right position in the visual field, which is called the receptive field of that cell. The receptive fields of two retinal ganglion cells, an 'on-centre' and an 'off-centre', are shown at the top of the figure. A cross indicates that light falling in this region increases the impulse rate from the cell, whereas light in the regions marked by minus signs slows the cell with a transient increase when it is extinguished. The next row shows the responses for centred spots and displaced spots for the two types. The responses consist of electrical impulses propagated along the nerve, and it will be seen that only one of the cells increased its rate of impulse firing when the light came on; the rate of the other actually decreased during the stimulus, but increased when the light went off. The situation is actually more complicated than this. If the stimulus spot had been displaced from the centre of the receptive field, or replaced by an annulus illuminating the region around the centre of the cell's receptive field, exactly the reverse pair of responses would have been recorded; the cell which increased in rate on illumination with a central spot would have slowed upon illumination in its annular surround, and speed up on extinction of this light, while the other cell would have done the reverse. Below this line, the effects of illuminating the whole of the central region, and illuminating the surround without the centre are shown, while the bottom part of the figure shows that illumination of both parts together is relatively ineffective; the two zones inhibit each other. The two types can be thought of as signalling local 'whiteness' or 'blackness', and the ineffectiveness of uniform illumination shows that they respond to local contrast, not the absolute level of illumination. Thus it may seem intuitively right that we have separate cells that signal 'whiteness' and 'blackness', the two types that respond as described above to increases and decreases in illumination. It also seems right that they respond to contrast rather than absolute illumination level, for we are all familiar with the fact that a level of luminance that looks white when surrounded by blackness can look black if the material surrounding it has its illumination greatly increased; in the former case 'on' cells would respond with increased firing, whereas in the latter it would be the 'off' cells.

This pattern of brief electrical impulses in two million optic nerve fibres is an obligatory stage in the representation of everything we see, intervening between the visual scene and our sensations of it. The retinal ganglion cells convert optical image signals filtered by the specific receptors of the retina into electrical impulses. In the cortex, the image is still represented by a pattern of impulses, but at any one time only a small proportion of the vast array of cortical neurones is active, because cortical neurones are more selective about what each of them responds to. As a result, when a cell does respond, it says something important and more specific about the image.

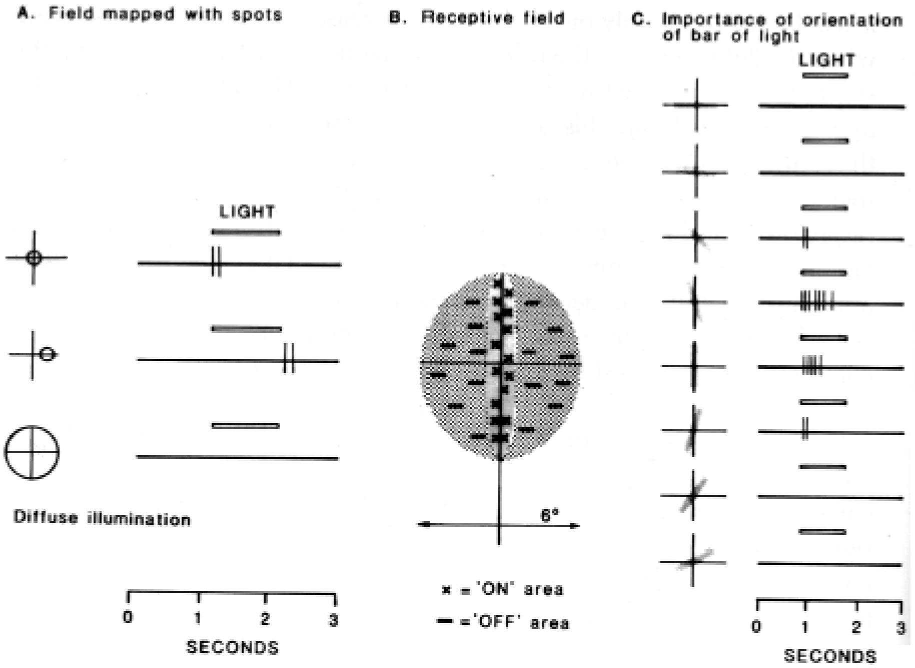


Figure 2.22: The responses of a neuron in the visual cortex

The receptive field of a neuron in primary visual cortex and its responses to oriented stimuli is shown in figure 2.22. These cells are usually responsive only if the stimulus is elongated and oriented correctly[23,24], as well as being positioned in the appropriate place in the visual field. The cell above for example responded best to a nearly vertical stimulus, but there is a full range of cells at each position in the visual field, each responding best to a different orientation, behaving like edge and pose detectors.

The circuitry of cerebral cortex contains for each square millimetre over 100,000 of those cells, and there are 100,000 square millimetres of cortex on each side of our brain. In the optic nerve, the image is carried in some million fibres from each eye and at any given moment, most of these fibres are carrying impulses at a rate between say 5 per second and their maximum of about 500 per second, which equals approximately 500 million compact feature information per second; 500MB/sec if each information piece were as simple as a computer bit[7].

This is an interesting way to represent an image, and the fact that each individual nerve cell conveys an important piece of information makes us feel that we have made some progress in understanding how images are 'digested'. But there is also something profoundly unsatisfactory about it: what earthly use is an array of 100 million cells, each of which responds to some rather specific characteristic of a small part of the visual field? The images we are familiar within our heads have a unity and usefulness that this representation, fragmented into a vast number of tiny pieces like a jigsaw puzzle, seems to lack. Why is it represented like this? Is it simply a parallel processing or error preventing structure? How is the picture on the jigsaw puzzle detected, or rather what neural mechanisms carry the analysis of the image further, and what is the goal of these further steps?

The brain must do more than 'see' images; it must also solve these neural problems.

2.6. Understanding the Understanding

The first step in understanding the understanding process must be to analyse the relationships of the pieces of an image to each other, to find the patterns and regularities it contains[18]. The links that show relationships within images were studied first by a group of psychologists, the Gestalt School[19]. The Gestalt school appreciated very clearly that it made no sense to consider images as a large number of separate fragments, and they were therefore much concerned with internal structure. They demonstrated that there are interactions between the parts of an image and described these in terms of principles such as 'grouping', 'good continuation', 'pragnantz', or 'common fate' as seen in the previous sections. Perhaps, because these principles were given a somewhat mystical status by their proponents, they were not very easily assimilated by many psychologists. But the attempts to perform visual tasks on computers now make it very clear what their role really is that they are the links within an image that the visual system uses, so they define the basis for image understanding.

If one considers how the Gestalt program of perception might be achieved one sees that there are two stages: first, local properties of an image such as colour, direction of movement, texture, or binocular disparity, must be detected; second, the information so gained must be re-assembled. We have learned in the previous sections that the pattern selectivity of cells in primary visual cortex fits in well with the view that they are performing the first operation, neurones respond selectively to the characteristics of the image. How can the second step of re-assembly be achieved?

2.7. Reassembly of the Feature Windows

Selective addressing and non-topographical neural images suggest[7,17,18] that reassembly is achieved by the cells in primary visual cortex, such as the pyramidal cell, grow into other cortical areas where they create new patterns in which the information is brought together according to new principles that are not necessarily related to the topography of the original image.

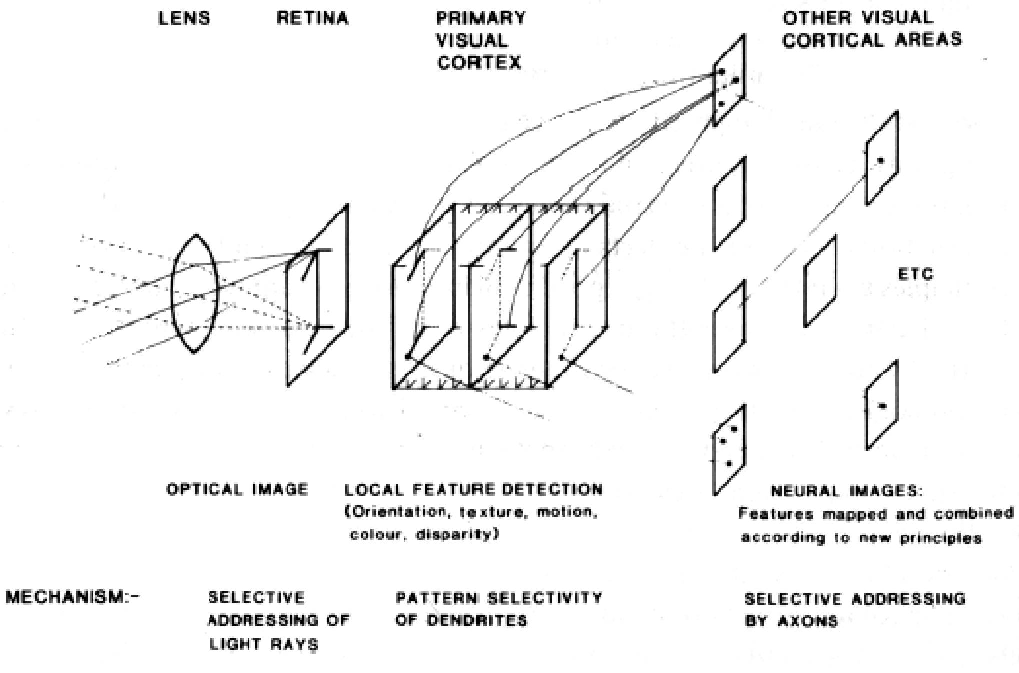

LENS
RETINA
PRIMARY VISUAL CORTEX
OTHER VISUAL CORTICAL AREAS
ETC
OPTICAL IMAGE
LOCAL FEATURE DETECTION (Orientation, texture, motion, colour, disparity)
NEURAL IMAGES: Features mapped and combined according to new principles
MECHANISM:-
SELECTIVE ADDRESSING OF LIGHT RAYS
PATTERN SELECTIVITY OF DENDRITES
SELECTIVE ADDRESSING BY AXONS

Figure 2.23: From optical to neural images

On the left of the figure above is the familiar ray diagram showing the formation of an optical image on the retina. Focussing by the lens ensures that light from a particular point on a distant object reaches a single position on the retina, and the necessary bending of the rays can be regarded as a form of selective addressing:

rays are bent upwards under one condition, downwards under another therefore direction of visual entry, a particular part of an object in the external world, is mapped to position in the image.

The central block depicts the many types of feature detector, represented as successive layers each dealing in parallel with the same image specialising in a different feature such as colour, depth, and motion, etc. Here, the three layers that respond selectively to the orientations of edges in the head, tail and shaft of the arrow image are shown. The information from each set of feature detectors is then projected onto another plane where similarity of features determines the position, not simply similarity of position in the original image.

It is hypothesised that the next step is the formation of many different neural images, in which features of the original image are mapped and brought together according to the principles suggested by Gestalt psychology, and perhaps others yet unknown. It is thought that cortical neurones achieve this by addressing their axons to positions that depend upon their own pattern selectivity, as well as their position in the image as exampled in the last two paragraphs. In this way, parts of the image can become related by their properties, as well as by their positions in the visual field.

2.8. Data Redundancy and Perception Success

The last issue that we want to observe on the human visual system is the feature reduction capability, the value estimation of the information[18,17,25]. Not all the 500MB/sec compact information follow the same route in the brain, some of them are reserved and used just only if they are required. This is one of the big difficulties that the current digital retrieval and puzzle problems face, redundancy of the information. The eyes and brain pair are careful searchers and selectors of important data:

Mervyn Thomas, Jane Mackworth[25] used a camera to motion pictures of the eye movements of drivers in actual traffic. They saw how the driver's eyes dart about in their search for information, when an automobile is moving, the driver’s eyes are constantly sampling the road ahead. At intervals he flicks quickly to the near curb, as if to monitor his position but for such monitoring he seems to rely chiefly on the streaming effect, the flow of blurred images past the edges of his field of vision. The edges of other vehicles and sudden gaps between them attract visual attention, as do signs along the roadside and large words printed on trucks. If something difficult to identify is encountered, the fixations are longer and the eyes jump back to view it again. The faster the automobile is moving, or the heavier the traffic, the more frequent are the saccades. When the driver stops at a traffic signal, his eyes seem to move less often and rather aimlessly, but they jump toward anything novel that appears. On a main highway the cars passing in the opposite direction attract quick glances. A broken white line along the centre of the road sometimes gives rise to vertical flicking movements. The eyes are also drawn to objects in the skyline such as tall buildings. One of the strongest visual attractions seems to be flashing lights, such as those of a turn indicator on a vehicle ahead or of a neon sign at the side of the road. This demonstrates an important characteristic of human vision. When the image of an object strikes the periphery of the retina, the eyes swing involuntarily so that the object is focused on the two foveas and can be perceived in detail. Although the periphery of the retina is poorly equipped for resolving detail, it is very sensitive to movement seen in the "corner of the eye." A flashing light therefore serves as a powerful visual stimulus. On several occasions during the experiments a driver continued to glance in the direction of the flashing indicators of a car ahead, even after it had made its turn.

Another example of peripheral attraction is observed when the eye movements of a pilot landing a small aircraft are recorded[25]. At touchdown the pilot usually maintained his sense of direction by the streaming effect while looking rather aimlessly ahead up the runway, this aimless looking reflects a readiness to react visually to the unexpected. On one occasion the pilot's eyes flicked away to fixate repeatedly on an object at the side of the runway in a flurry of rapid saccades. His eyes continued to be drawn over even after he must have identified the object as one of the spruce seedlings used on that airfield as snow markers, which our record showed he was fixating accurately. This sensitivity to a moving or novel object at the edge of the scene demonstrates that the retina functions as an effective wide-angle early-warning system that a strong peripheral, signal will continue to pull the eyes. This is the objective of the designer of flashing neon signs.

In reading, as in driving an automobile, the predominant eye movement is the saccade, but the saccade of reading is initiated in different way, when one gazes at a line on a printed page, only three or four words can be seen distinctly. If every word in the line is to be read, the eyes must jump two or three times. How often they jump depends not only on the reader's ability to process the visual information but also on his interest in what he is reading. Thus the reading saccade is initiated not so much by the image on the periphery of the retina as by a decision made within the central nervous system. Fixation times lengthen as the material becomes harder to comprehend. The eyes may return at intervals to words scanned earlier; these regressions indicate the time it has taken the reader to recognize that his processing of the information was incomplete or faulty.

The human vision is not always perfect. This is one reason why we try to computerise the content retrieval. Edward L. Lansdown[25] recorded the eye movements of a group of student radiologists as they inspected a selection of chest X-rays. The records showed that the students had carefully examined the edges of the heart and the margins of the lung fields, and indeed these are important regions for signs of disease. But large areas of the lung fields were never inspected by most of the students in the group, even though they thought they had scanned the films adequately. To be sure, the students who had made the most complete visual examinations were the ones with the most experience in X-ray interpretation. William J. Tuddenham[25] of the University of Pennsylvania School of Medicine and L. Henry Garland of the Stanford University School of Medicine tested groups of trained radiologists and found that they missed 25 to 30 percent of "positive" chest X-rays under conditions in which their errors must have been largely due to failures of perception. Joseph Zeidner and Robert Sadacca of the Human Factors Research Branch of the U.S. Army[25] have reported similar failures in the interpretation of aerial photographs by a group of skilled military photo-interpreters, they neglected to report 54 percent of the significant signs. It appears that the structure of the image under examination may obliterate the pattern of scanning an

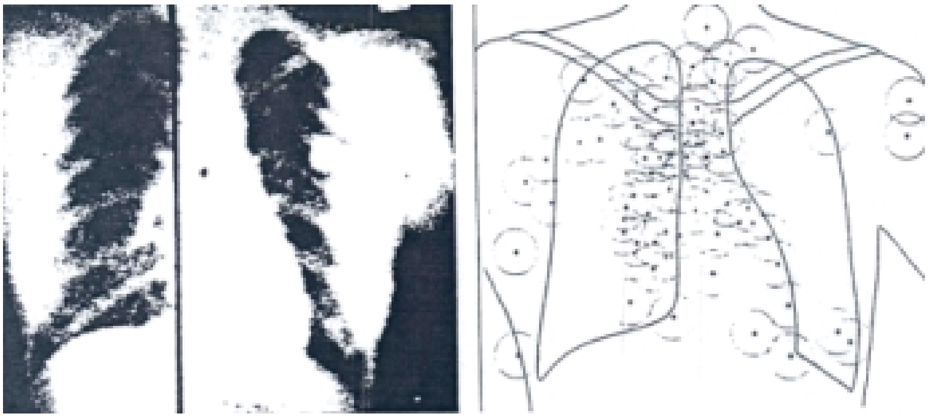


Figure 2.24: Performance of the human vision

observer intends to follow; his gaze is drawn away, so that he literally overlooks areas he believes he has scanned. Moreover, low-resolution peripheral vision often determines where the viewer does not look.

There are also interesting differences of perception between people: Eight percent of women see an extra hue component of the colour[7]; N. Mackworth working at the Centre for Cognitive Studies at Harvard University[25], have found that children show more short eye movements and concentrate more on less informative details.

A great deal of the information that arrives at the brain from the retina fails to obtrude on the consciousness. In this connection it is startling to watch a film of one's own eye movements. The record[25] shows hundreds of fixations in which items were observed of which one has not the slightest recollection. Yet the signals must have reached the brain because one took motor action and even made rather complex decisions based on the information that was received during the forgotten fixation. Parts of the brain appear to function rather like a secretary who handles routine matters without consulting her employer and apprises him of important points in incoming letters but who at times makes mistakes.

The link between the image and the mind is a difficult one to investigate because the brain does not receive information passively but partly controls what reaches it from the eyes. As Darwin constructed his famous evolution theory, he was obstructed with the complexity of the eye and called the structure as 'cold shudder'[6]. The processes in brain related to vision is really like a black box; contains valuable information for our digital problems but not fully understood yet, every single observation advances us just a step above.

In concluding the biological vision chapter, we would like to emphasise that the feature extraction step is just a single face of the content retrieval; proper feature spaces, distance metrics, similarity analysis techniques, memory behaviours, indexing, semantic association and the help of the other branches are required to handle such a complex job. In the following chapters, we'll see that actually there is lots of information on how to derive local features with performances quite similar to the human eye. The reason why we couldn't realise robust content based retrieval systems and couldn't solve puzzle problems as powerful as the human eye may be the ignorance on how the brain reassembles the pieces again using these local features, at semantic behaviours.

CHAPTER 3

CONTENT BASED IMAGE RETRIEVAL SYSTEMS

In this chapter, we’ll cover the content-based systems. We first start with the needs behind the multimedia databases and then summarise the content-based systems available in the literature. After listing the application areas and successful platform examples, we focus on the low level feature systems with their similarity metrics and performance criteria.

3.1. The Growth of Digital Multimedia Data

The use of images in human communication is hardly new, our cave-dwelling ancestors painted pictures on the walls of their caves, and the use of maps and building plans to convey information almost certainly dates back to pre-roman times. Technology, in the form of inventions such as photography and television, has played a major role in facilitating the capture and communication of image data. But the real engine of the imaging revolution has been the computer, bringing with it a range of techniques for digital image capture, processing, storage and transmission. The involvement of computers in imaging can be dated back to 1965, with Ivan Sutherland's *Sketchpad* project, which demonstrated the feasibility of computerised creation, manipulation and storage of images, though the high cost of hardware limited their use until the mid-1980s[56]. As large amounts of both internal and external memory become increasingly less expensive and processors become increasingly more powerful, image databases have gone from an expectation to a firm reality[1]. Image production and use now routinely occurs across a broad range of disciplines and subject fields like art galleries and museum management, architectural and engineering design, interior design, remote sensing and earth resource management, geographic information systems, scientific database management, weather forecasting, retailing, fabric and fashion design, law enforcement and criminal investigation, picture archiving and communication systems. The creation of the worldwide web in the early 1990s, enabling users to access data in a variety of media from anywhere on the planet, has provided a further massive stimulus to the exploitation of digital images. The number of images available on the Web was estimated to be between 10 and 30 million in 1997 by Sclaroff et al.[55]

3.2. From Databases to Visual Information Retrieval Systems

The need to find the desired image from a collection is shared[4,1,55,56,62] by many professional groups in domains such as crime prevention, medicine, architecture, art, fashion, publishing etc. Uses vary according to application: art collection users may wish to find a work of art by a certain artist or to find out who painted a particular image they have seen. Medical database users may be medical students studying anatomy or doctors looking for sample instances of a given disease.

Image databases can be huge, containing hundreds of thousands or millions of images. Based on initial calculations, it was estimated that a single 14"x17" radiography could be digitised by 24MB approximately depending on the resolution. This combined with other factors such as an examination requiring several radiographs, and the average number of examinations per year left results that an image database for a typical 500-bed hospital would require at least 15 terabytes storage per year[1]. In most cases, these databases are only indexed by key words that have to be decided upon and entered into the database system by a human categoriser. Titles, authors, captions, and descriptive labels provide a natural means of summarising massive quantities of information. First-generation visual information retrieval systems allowed the access to images through these string attributes, which summarise in words what is represented in and its meaning. Text annotation consumes little space and provide fast retrieval of large amounts of data with the help of traditional search engines working in the textual domain either using traditional query languages, like SQL, or full text retrieval based on natural language processing and artificial intelligence methods. But, when an image database needs annotation, a person must enter the labels by hand at great cost and tedium caused by the subjective, individual interpretation of the 'non-verbal symbolism' of an image. The basic problem of the first generation systems is what the picture means, or whether it means anything at all cannot be clearly stated with any certainty.

3.3. New-generation Visual Information Retrieval Systems

New-generation visual information retrieval systems[4] support full retrieval by visual content. Access to visual information is not only performed at a conceptual level, using keywords as in the textual domain, but also at a perceptual level, using objective measurements of the visual content and appropriate similarity models. In these systems, image processing, pattern recognition and computer vision are an integral part of the system's architecture and operation. They permit the objective analysis of pixel distribution and the automatic extraction of measurements from raw sensory input.

3.3.1. Content Based Image Retrieval (CBIR) systems

The earliest use of the term *content-based image retrieval* in the literature seems to have been by Kato in 1992, to describe his experiments into automatic retrieval of images from a database by colour and shape features[4,1]. The term has since been widely used to describe the process of retrieving desired images from a large collection on the basis of features such as colour, texture and shape that can be automatically extracted from the images themselves. The features used for retrieval can be either primitive or semantic, but the extraction process must be predominantly automatic. Retrieval of images by manually assigned keywords is definitely not CBIR as the term is generally understood, even if the keywords describe image content.

Alphanumeric databases allow a large amount of data to be stored in a local repository and accessed by content through appropriate query languages; Information is structured so as to ensure efficiency. On the other hand, CBIR systems have provided access to unstructured textual documents since digitised images consist purely of arrays of pixel intensities, with no inherent meaning. One of the key issues with any kind of image processing is the need to extract useful information from the raw data, such as recognising the presence of particular shapes, colours or textures before any kind of reasoning about the image's contents is possible. Image databases thus differ fundamentally from alphanumeric databases, where the raw material, words stored as ASCII character strings, have already been logically structured by the author.

Content-based image retrieval involves a direct matching operation between a query image and a database of stored images. The process involves computing a feature vector for the unique characteristics of the image. Similarity is computed by comparing the feature vectors of the images. The result of this process is a quantified similarity score that measures the visual distance between the two images represented by the feature vectors.

The problems of image retrieval in image encoding, storage, compression, transmission, display, feature description and matching have been widely recognized, and the search for solutions becomes an increasingly active area for research and development. Some indication of the rate of increase can be gained from the number of journal articles appearing each year on the subject, growing from 4 in 1991 to 12 in 1994, and 45 in 1998[56].

The variety of knowledge required in visual information retrieval is large. Different research fields, which have evolved separately, provide valuable contributions to this new research subject. Information retrieval, visual data modelling and representation, image/video analysis and processing, pattern recognition and computer vision, multimedia database organisation, multi dimensional indexing, psychological modelling of user behaviour, man-machine interaction and data visualisation, software engineering are only the most important research fields that contribute in a separate but interrelated way to visual information retrieval.

3.3.2.Characteristics of Data Types

In visual information retrieval, two different types of information are associated[4] with images:

The first type of data is not directly concerned with image/video content, but in some way related to it; it is also referred to as content-independent metadata. Examples of such data are the format, the author’s name, date, location, ownership, etc. Along the thesis, a visual content searcher that works on the Internet is developed, which gets a code page, parse the image tags, download the content and finally iterates this loop until information request is satisfied. We see the content-independent data queries at many CBIR systems but as stated before, the importance of the CBIR systems originates from image features explained below.

The second type of data refers to the visual content of images and has two levels. The first level of the data refers to low/intermediate-level features, like colour, texture, shape, spatial relationship, motion, and their combinations. It is also referred as content-dependent metadata and generally, this data is concerned with perceptual facts. For still images, features immediately perceived are colour and texture; this is the data class we focused on this thesis. The higher level of data may refer to content semantics and referred to as content-descriptive metadata. It is concerned with relationships of image entities with real-world entities or temporal events, emotions and meaning associated with visual signs and scenes.

The type of data used to access images and characteristics of image queries have a direct impact on the internal organisation of the retrieval system, on the way in which retrieval is performed and, naturally, on its effectiveness. Another important factor for the system design is the characteristics of image queries.

3.3.3. Characteristics of Image Queries

What kinds of query are users likely to put to an image database? Query types may be classified[55,56] into three levels of increasing complexity, which is closely related with the data types described above:

Level 1 comprises retrieval by *primitive* features such as color, texture, shape or the spatial location of image elements. Examples of such queries might include "find pictures with long thin dark objects in the top left-hand corner", "find images containing yellow stars arranged in a ring" - or most commonly "find me more pictures that look like this". This level of retrieval uses features, such as a given shade of yellow, which are both objective, and directly derivable from the images themselves, without the need to refer to any external knowledge base. A typical system allows users to formulate queries by submitting an example of the type of image being sought. The system then identifies those stored images whose feature values match those of the query most closely, and displays thumbnails of these images on the screen. Its use is largely limited to specialist applications such as identification of drawings in a design archive, or color matching of fashion accessories.

All current CBIR systems, whether commercial or experimental, operate at level 1. We are also interested in level 1 systems, which will be studied in detail in the following sections.

Level 2 comprises retrieval by *derived,* sometimes known as *logical or* semantic features, involving some degree of logical inference about the identity of the objects depicted in the image. It can usefully be divided further into:

i. Retrieval of objects of a given type, e.g. "find pictures of a double-decker bus";

ii. Retrieval of individual objects or persons, e.g. "find a picture of the Eiffel tower".

To answer queries at this level, reference to some outside store of knowledge is normally required. In the first example above, some prior understanding is necessary to identify an object as a bus rather than a lorry; in the second example, one needs the knowledge that a given individual structure has been given the name "the eiffel tower". Search criteria at this level are usually still reasonably objective. This level of query is more generally encountered than level 1, for example, most queries received by newspaper picture libraries appear to fall into this overall category.

Level 3 comprises retrieval by *abstract* attributes, involving a significant amount of high-level reasoning about the meaning and purpose of the objects or scenes depicted. Again, this level of retrieval can usefully be subdivided into:

i. Retrieval of named events or types of activity (e.g. "find pictures of Scottish folk dancing");

ii. Retrieval of pictures with emotional or religious significance ("find a picture depicting suffering").

Success in answering queries at this level can require some sophistication on the part of the searcher. Complex reasoning and often subjective judgement can be required to make the link between image content and the abstract concepts it is required to illustrate.

Reports of automatic image retrieval at level 3 are very rare. The only research that falls even remotely into this category has attempted to use the subjective connotations of color such as whether a color is perceived to be warm or cold, or whether two colors go well with each other to allow retrieval of images evoking a particular mood.

Together with the biological vision chapter, this classification of query types illustrates the limitations of current image retrieval techniques. The most significant gap at present lies between levels 1 and 2. The levels 2 and 3 together is referred as semantic image retrieval, and hence the gap between levels 1 and 2 as the semantic gap.

The system designed in this thesis focuses totally on the low level features; but surely, as a next step, semantic associations should be added.

3.3.4 Practical Applications of CBIR

There are many applications[4,1,55,56] where full visual retrieval of still images and videos is important. However, despite the fact that the interest in this topic is growing fast, its application in real contexts is in its infancy. Promising fields of application for still images include:

Crime prevention: Law enforcement agencies typically maintain large archives of visual evidence, including past suspects' facial photographs, fingerprints, and shoeprints. Whenever a serious crime is committed, they can compare evidence from the scene of the crime for its similarity to records in their archives or verify the identity of a known individual and those capable of searching an entire database to find the closest matching records.

The military: Military applications of imaging technology are probably the best developed; though least publicized. Recognition of enemy aircraft from radar screens, identification of targets from satellite photographs, and provision of guidance systems for cruise missiles are known examples

Architectural and engineering design: Architectural and engineering design share a number of common features. The use of stylized 2- and 3-D models to represent design objects, the need to visualize designs for the benefit of non-technical clients, and the need to work within externally imposed constraints, often financial, constrain the designer to be aware of previous designs. Hence the ability to search design archives for previous examples which are in some way similar, or meet specified suitability criteria, can be valuable.

Fashion and interior design: Similarities can also be observed in the design process in other fields, including fashion and interior design. Here again, the designer has to work within externally imposed constraints, such as choice of materials. The ability to search a collection of fabrics to find a particular combination of color or texture is increasingly being recognized as a useful aid to the design process.

Journalism and advertising: Both newspapers and stock shot agencies maintain archives of still photographs to illustrate articles or advertising copy. These archives can often be extremely large, running into millions of images, and dauntingly expensive to maintain if detailed keyword indexing is provided. Broadcasting corporations are faced with an even bigger problem, having to deal with millions of hours of archive video footage, which are almost impossible to annotate without some degree of automatic assistance. CBIR techniques can be used to break up a video sequence into individual shots, and generate representative keyframes for each shot. It is therefore possible to generate a storyboard for each video entirely by automatic means. This application area is probably one of the prime users of CBIR technology at present.

Medical diagnosis: The increasing reliance of modern medicine on diagnostic techniques such as radiology, histopathology, and computerized tomography has resulted in an explosion in the number and importance of medical images now stored by most hospitals. While the prime requirement for medical imaging systems is to be able to display images relating to a named patient, there is increasing interest in the use of CBIR techniques to aid diagnosis by identifying similar past cases.

Examples of this include the I^2C system for retrieving 2-D radiological images from the University of Crete, and the 3-D neurological image retrieval system being developed at Carnegie-Mellon University, both developed with the aim of assisting medical staff in diagnosing brain tumors.

Geographical information systems (GIS) and remote sensing: Although not strictly a case of image retrieval, managers responsible for planning marketing and distribution in large corporations need to be able to search by spatial attribute, e.g. to find the 10 retail outlets closest to a given warehouse. And the military are not the only group interested in analyzing satellite images. Agriculturists and physical geographers use such images extensively, both in research and for more practical purposes, such as identifying areas where crops are diseased or lacking in nutrients or alerting governments to farmers growing crops on land they have been paid to leave lying fallow.

Cultural heritage: Museums and art galleries deal in inherently visual objects. The ability to identify objects sharing some aspect of visual similarity can be useful both to researchers trying to trace historical influences, and to art lovers looking for further examples of paintings or sculptures appealing to their taste.

IBM's QBIC system has received extensive trials in managing art library databases, and has proved an extremely useful browsing tool. Jain et al has applied CBIR techniques to the management of image and video data relating to a Hindu temple in India.

Education and training: It is often difficult to identify good teaching material to illustrate key points in a lecture or self-study module. The availability of searchable collections of video clips providing examples of avalanches for a lecture on mountain safety, or traffic congestion for a course on urban planning, could reduce preparation time and lead to improved teaching quality.

Home entertainment: Much home entertainment is image or video-based, including holiday snapshots, home videos and scenes from favorite TV programs or films. This is one of the few areas where a mass market for CBIR technology could develop.

Web searching: Cutting across many of the above application areas is the need for effective location of both text and images on the Web, which has developed over the last five years into an indispensable source of both information and entertainment. Text-based search engines have grown rapidly in usage as the Web has expanded; the well-publicized difficulty of locating images on the Web indicates that there is a clear need for image search tools of similar power.

Two commercial Web search engines now offer a CBIR option, the Yahoo! Image Surfer, based on Excalibur technology and AltaVista's AV Photo Finder, using Virage technology.

3.3.5. Available CBIR Softwares

Despite the shortcomings of current CBIR technology, several image retrieval systems, are now available as commercial packages[4,1,55,56,62]. Some of the most prominent of these are described below:

QBIC: IBM's QBIC system is probably the best-known of all image content retrieval systems. It offers retrieval by any combination of color, texture or shape, as well as by text keyword. Image queries can be formulated by selection from a palette, specifying an example query image, or sketching a desired shape on the screen. The system extracts and stores color, shape and texture features from each image added to the database. For still images, functions available are querying by semantic content, by global color similarity, by color-region similarity, by texture similarity; by shape and spatial relationship similarity. At search time, the system matches appropriate features from query and stored images, calculates a similarity score between the query and each stored image examined, and displays the most similar images on the screen as thumbnails.

Virage: Another well-known commercial system is the VIR Image Engine from Virage, Inc. This is available as a series of independent modules, which systems developers can build into their own programs. This makes it easy to extend the system by building in new types of query interface, or additional customized modules to process specialized collections of images. For still images, functions available are querying by semantic content, by global colour similarity, by texture similarity, by structure similarity.

Excalibur: A similar philosophy has been adopted by Excalibur Technologies. It is marketed principally as an applications development tool rather then as a standalone retrieval package. Its best-known application is probably the Yahoo! Image Surfer, allowing content-based retrieval of images from the Worldwide Web.

Visual Retrievalware is a framework for retrieval of still images only. Functions available are querying by semantic content; by global color similarity, by texture similarity.

Experimental systems: A large number of experimental systems have been developed, by academic institutions, in order to demonstrate the feasibility of new techniques. Some of the best-known are the Photobook system, from Massachusetts Institute of Technology, The VisualSEEk system at Columbia University, the Surfimage system from INRIA, France.

CBIR at present is still very much a research topic. The technology is exciting but immature, and few operational image archives have yet shown any serious interest in adoption. The application areas most likely to benefit from the adoption of CBIR are those where level 1 techniques can be directly used. Areas where retrieval by primitive image feature is likely to be beneficial are crime prevention including identification of faces and fingerprints, architectural design to retrieve of similar previous designs, and medical diagnosis, retrieval of cases with similar features. It is also unlikely that general-purpose image retrieval software will meet the needs of these user communities without a significant degree of customization. Each of these application areas has their own range of special needs and constraints. Software solutions that fail to address these needs are unlikely to perform well enough to convince users that they are worth adopting.

CBIR systems are also not particularly easy for inexperienced end-users to understand. It is certainly not obvious to the casual user how to formulate and refine queries couched in terms of color, texture or shape features. There is thus an argument for the involvement of an experienced search intermediary who can translate a user's query into appropriate image primitives, and refine the search in consultation with the user in the light of output received. This kind of role is less fanciful than it might seem that it is simply trying to exploit the same approach as some researchers into semantic image retrieval. The only difference is that it would use humans instead of machines, and therefore probably has a higher chance of success. Such intermediaries would be difficult to train because the requisite body of knowledge exists only in fragmented form, if at all. But they could be enormously helpful in making CBIR systems usable.

In the next section, we’ll start to explore the level 1 static image retrieval systems in detail.

3.4. Level 1 Content Based Static Image Retrieval Systems

Perceptual content of still images includes colour, texture, shape and spatial relationship. To retrieve images according to perceptual properties, the basic retrieval paradigm requires[4,1] that, for each image, a set of distinguishing features, i.e. model parameters are precomputed. Queries are expressed through visual examples. To initiate a query, the user selects which features and model parameter ranges are important and chooses a similarity measure. Examples can either be authored by the user or extracted from image samples. The system checks the similarity between the visual content of the user's query and the database images. Since one cannot expect results obtained in response to a query to be fully satisfactory, the commonly followed technique to improve the quality of retrieval is to keep as low as possible the number of misses at the expense of a larger number of false retrievals. Similarity-based retrieval differs in two basic respects from matching:

Matching, as it is commonly defined in computer vision for object recognition, is a binary partition operation, being intrinsically committed to deciding whether or not the object observed corresponds to a model.

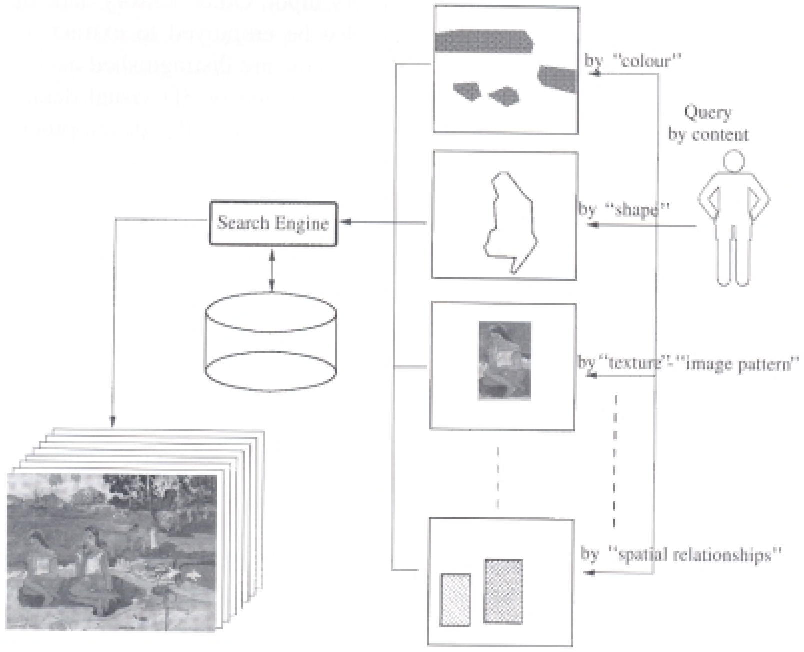


Figure 3.1: Different types of queries[4]

Similarity-based retrieval is, instead, the task of re-ordering database images according to their measured similarity to a query example. It is therefore concerned with ranking rather than classification. It does not postulate the existence of a target image. Re-ordering of database images is performed, even if there is no image close to the example.

In matching, uncertainties and imprecision are commonly managed during the process. Features used to perform classification are chosen according to the problem. The way in which matching is carried out depends on the description adopted.

In retrieval by similarity, the user is in the retrieval loop. Therefore, it is he or she who defines a query, analyses system responses and refines the query. The existence of the user in the retrieval loop underlines instead the importance of flexible interfaces and visualization tools.

Similarity-based retrieval requires that feature matching is robust with respect to differences between the query and the images. Similarity models should somehow replicate the way in which humans assess similarity between different objects. This approach is complicated by the fact that, unfortunately, there is not a single model of similarity. Moreover, often the user combines different similarity measures, each specialised on a specific domain, so as to derive a subjective measure suited for the context of operation.

3.4.1. Similarity Metric Models

Pre-attentive and attentive human similarities differ from each other[4,10,1]: Attentive similarity has to do with interpretation and involves previous knowledge, a form of reasoning. Pre-attentive similarity instead is simply based on the perceived similarity between stimuli, with no form of interpretation.

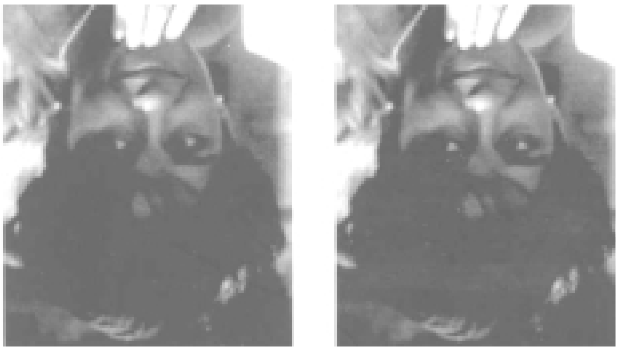


Figure 3.2: Pre-attentive and attentive similarity test

Figure shows an example of such a distinction. The two faces are in an unfamiliar position but, at first glance, seem to be the same. Turning the page upside down, so that the faces are in a more familiar position, the face images are interpreted and the differences in eye contour can be realised. In the first case, pre-attentive similarity is involved. Features considered are those prominent in the image, like colour, texture, presence and location of salient elements of a face such as eyes, mouth, nose; whereas, the precise identification of these elements and their recognition through feature matching are tasks performed when attentive similarity is involved. Attentive similarity is requested in domain-specific retrieval applications such as facial, iris, mechanical-part or medical databases. These use object matching and ad-hoc similarity criteria, particular to the application domain.

Pre-attentive similarity, instead, is requested in retrieval applications where colour, texture, shape and spatial relationship perception is more important. These need similarity models that closely conform to human similarity perception of censorial stimuli. Some of these models are briefly summarised in the following paragraphs because our concern is mainly to focus on pre-attentive similarity:

A very well known theory[4] postulates that human similarity perception is based on the measurement of an appropriate distance in a metric psychological space. In this theory, it is assumed that a set of features model the stimulus' properties so that it can be represented as a point in a suitable feature space. If d is a distance function and S_1, S_2 and S_3 are generic stimuli, the following metric axioms must be verified:

. Constancy of self-similarity:

$$d(s_1,s_1)=d(s_2,s_2)=0; \tag{3.1}$$

. Minimality:

$$d(s_1,s_2) \geq d(s_1,s_1) \geq 0; \tag{3.2}$$

. Symmetry property:

$$d(s_1,s_2)=d(s_2,s_1); \tag{3.3}$$

. Triangle inequality property:

$$d(s_1,s_2)+ d(s_2,s_3) \geq d(s_1,s_3) \tag{3.4}$$

Obeying the axioms above, the commonly used distance functions[4,1,10,55] of the content similarity analysis are:

The Euclidean distance:

$$d(s_1,s_2)=\sqrt{(x_2-x_1)^2+(y_2-y_1)^2} \qquad (3.5)$$

The city-block distance:

$$d(s_1,s_2)=\| x_2-x_1 \| + \| y_2-y_1 \| \qquad (3.6)$$

The Minkowsky distance:

$$d(s_1,s_2)=\sum_{k=1}^{n} | x_k - y_k | \qquad (3.7)$$

where X_k, Y_k are generic features of the stimulus S_k.

Many studies in psychology have pointed out certain inadequacies of the feature vector model and metric distances. For example, for shape similarity, feature vectors, even of high dimensionality and metric distances such as the Euclidean do not result as being close to the human judgement of similarity.

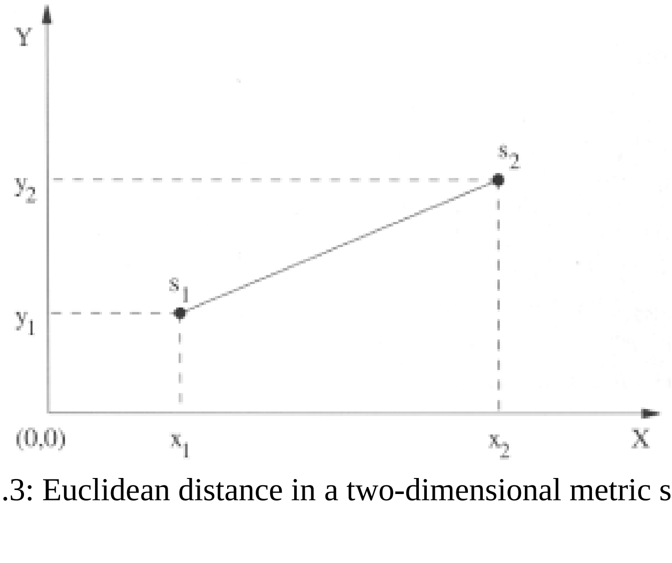


Figure 3.3: Euclidean distance in a two-dimensional metric space

Nevertheless, this model also has several advantages that make it the most widely employed in visual information retrieval systems:

I. It is adequate for certain types of human similarity judgement. For example, similarity judgement of colour stimuli is well approximated through this model.

II. It is consistent with feature-based description, which is a well-established approach in pattern recognition and computer vision. A lot of feature extraction algorithms are available.

III. If visual data are modelled through a feature vector, an index can be built for these vectors according to classical multi dimensional access methods. Indexes are very helpful when the number of database images is very large.

As an alternative method to the standard distance metrics, we applied the Mahalanobis distance, which is dominantly used in machine vision. The quantity r in

$$r^2 = (x - m_x)' C_x^{-1} (x - m_x) \qquad (3.8)$$

is called the Mahalanobis distance from the feature vector x to the mean vector m_x, where C_x is the covariance matrix for x. It can be shown that the surfaces on which r is constant are ellipsoids that are centered about the mean m_x. In the special case where the features are uncorrelated and the variances in all directions are the same, these surfaces are spheres, and the Mahalanobis distance becomes equivalent to the Euclidean distance. The use of the Mahalanobis metric removes several of the limitations of the Euclidean metric:

I. It automatically accounts for the scaling of the coordinate axes

II. It corrects for correlation between the different features

III. It can provide curved as well as linear decision boundaries

However, there is a price to be paid for these advantages. The covariance matrices can be hard to determine accurately, and the memory and time requirements grow quadratically rather than linearly with the number of features. These problems may be insignificant when only a few features are needed, but they can become quite serious when the number of features becomes large.

Finally, we'll mention about the texture signatures, which frequently used by the professional CBIR systems.

Image signatures are obtained by evaluating probability density functions over the multi-dimensional space of feature vectors. These are estimated as a weighted sum of gaussians:

$$S_I(x) = \sum_{(x,y)}^{k} w_i G_i(x) \tag{3.9}$$

$$G_i(x) = (2\pi)^{N/2} \exp\left(-1/2(x-\mu_i)^t \sum\nolimits_i^{-1}(x-\mu_i)\right) \tag{3.10}$$

Each gaussian G_i corresponds to a cluster of feature vectors. Clusters are obtained through a k-means clustering algorithm followed (optionally) by cluster merging. Mean vectors μ and covariance matrices Σ are derived for each Gaussian. Weights w_i of the gaussians are the number of elements in the cluster.

The similarity between query (I_Q) and target (I_T) textures is evaluated by comparing image signatures according to the following distance function:

$$M(I_Q, I_T) = \left(\int_R (S_Q(x) - S_T(x))^2 dx\right)^{1/2} \tag{3.11}$$

3.4.2. Standard Performance Evaluation Criteria

In traditional information retrieval, evaluation is performed[10,1] according to the following table:

	Judgement by evaluator	
	Relevant	Not relevant
Retrieved	*A (correctly retrieved)*	*B (falsely retrieved)*
Not retrieved	*C (missed)*	*D (correctly rejected)*

Table 3.1: Traditional metrics for information retrieval

Traditional performance evaluation metrics are 'recall' and 'precision'. They are a function of both correct matches and the relevance of a document to a query. Recall and precision have also been used for visual information retrieval. A more sophisticated measure of performance takes into account the effectiveness of retrieval.

It is a measure of the agreement between human evaluators and the system in ranking retrieved images according to their similarity to a query.

Recall measures the ability of the system to retrieve all documents that are relevant. It is defined as:

$$recall = \frac{relevant \quad correctly \quad retrieved}{all \quad relevant} = \frac{A}{A+C} \tag{3.12}$$

Precision measures the ability of the system to retrieve only documents that are relevant. It is defined as:

$$precision = \frac{relevant \quad correctly \quad retrieved}{all \quad retrieved} = \frac{A}{A+B} \tag{3.13}$$

Recall and precision require a ground truth to assess the relevance of images for a set of significant queries. Test sets are usually large but only a small fraction of relevant images is included. The relevance of a retrieved image is measured as the "perceived contact" between the visual content of a query and that of the image.

In contrast to recall and precision, measuring effectiveness requires a small image test-set (20-40). Images include visual elements such as textures, shapes, images that are somewhat different from the query example, yet representing elements of the same category as the query. For each image i of the test-set, and each query example j, a window can be used of width $\sigma_j(i)$ centred in the rank $P_j(i)$, which is assigned by the system. A measure of the difference between the system and human ranking can be represented by the sum of the percentage of people $Q_j(i,k)$ who ranked the ith image in a position between $P_j(i)$- $\sigma_j(i)$ and $P_j(i) + \sigma_j(i)$:

$$S_j(i) = \sum_{k=P_j(i)-\sigma_j(i)}^{k=P_j(i)+\sigma_j(i)} Q_j(i,k) \tag{3.14}$$

Computational complexity analysis is also important to check the possibility of real-time implementation which usually considers the most significant operations needed to resolve a typical query. As explained before, there is another field of image retrieval which focuses mainly on the computation time and efficient architectures. Our target is to create an offline application by evaluating the textural features. Therefore, this measure is not in our priority.

3.4.3. The Performance Evaluation Metrics of the Thesis

Since visual information retrieval is context dependent, we want to summarise below the performance measures that will be used in the thesis. We take two different reports from a run, a visual and a file report. The file version includes a complete similarity order report including all images' ranks for all possible requests. We define four images taken from the same object as relevant. The visual interface aims to show how correct the expected places of all the relevant images are filled, therefore only the closest 3 retrievals are displayed, which can be expanded when need. Such a definition decreases the success rate because different objects may have relevant primitive relations but at least doesn't require human subjective tests.

Precision-Recall: We keep "all retrieved" numbers fixed; For this reason, in our system the precision and recall metrics are the same. This new measure shows us how correct we call images on the top three places in rank.

The distribution of the shortest interval length for the closest relevant retrieval: This measure defines the length of estimated relevant region. It is the answer to the question how many images we should request to satisfy a success.

The distribution of the shortest interval length for all relevant retrieval: As stated before, these systems are reordering machines. It is not always possible, also there is no need, to get all images from a unique object at the top ranks. Sometimes, the inter-class relationships may be weaker than the intra-class ones. The texture feature is especially open to this case because of its scale dependency. So the images from the same object may be ordered at different places. This measure enables us to see the distribution of the relevant images over the database.

The percentage of the correctness for the closest retrieval: The closest retrieval's correctness is especially important for machine vision applications. Although we establish a similarity analysis, which doesn't need such strict decisions, choosing one of the relevant images as closest retrieval is a high quality performance measure.

The percentage of the all-first three correct retrieval: For any request, if the first three retrievals are relevant, then the success rate is optimal because we have only four samples from each class. Therefore, the ‘all-first three correct retrieval’ measure denotes the percentage of perfect results.

The precision metric by majority voting rule: In pattern analysis, the voting rule is a popular method to combine the results of different classifiers. In the similarity analysis, a final decision may be made by a voting rule among the first three retrievals to extract the performance. As a result, we can tolerate to some non-relevant retrievals without changing the class decided.

Now, we think that we are equipped enough to continue the topic by specialising on the ‘texture’ and ‘colour’ similarity retrievals.

3.4.4. Image Retrieval by Colour and Texture Similarity

Colour and texture are powerful discriminating features, present almost everywhere in nature. For retrieving images on the basis of colour similarity, several methods have been described in the literature[4,10,1,13], but most are variations of the same simple idea, colour histograms of chromatic content. When a user wants to find an image similar to a query image, his interest lies on the global image chromatic content. Colour similarity measures compare the colour content of one image with the colour content of a second image or of a query specification. Each image added to the collection is analysed to compute a *colour histogram* which shows the proportion of pixels of each colour within the image. The colour histogram for each image is then stored in the database. At search time, the user can either specify the desired proportion of each colour, 75% olive green and 25% red, for example, or submit an example image from which a colour histogram is calculated. Either way, the matching process then retrieves those images whose colour histograms match those of the query most closely, i.e. , the relative number of pixels that are in the histogram bin closest to the colour in a query. The matching technique most commonly used, histogram intersection, was first developed by Swain and Ballard in 1991. Variants of this technique are now used in a high proportion of current CBIR systems. We’ll state three major colour algorithms in the next chapter.

The ability to retrieve images on the basis of texture similarity can also be useful, especially for distinguishing between areas of images with similar colour such as sky and sea, or leaves and grass. Besides, textural features are more robust against light and environmental changes if the contrast measures are preferred. A variety of techniques have been used for measuring texture similarity; textures may be described according to their spatial, frequency or perceptual properties. Periodicity, coarseness, preferred direction, degree of complexity are some of the most perceptually salient attributes of a texture.

Texture queries can be formulated in a similar manner to colour queries, by selecting examples of desired textures from a palette, or by supplying an example query image. The system then retrieves images with texture measures most similar in value to the query.

In the following chapter, we’ll study the colour and textural feature extraction algorithms for content retrieval in detail.

CHAPTER 4

TEXTURE ANALYSIS

This chapter aims to build the background knowledge on texture analysis. First, a taxonomy discussion will be made. Then, texture methods will be categorised and studied in detail including colour algorithms.

There are many fields of image processing in which texture and colour play important roles. The most important areas are probably classification, image segmentation, image encoding and computer graphics. The classification of textures, often real textures, is a common problem in medical image processing and in process control. Some typical applications are the recognition of tissues in microscope images, the quality control of timber, line boards and paper, or the classification of reconnaissance and remote sensing images. Image segmentation is related to the classification problem; when other features, like shape, tone or colour, are not sufficient to discriminate between regions, one criteria of image segmentation might be texture.

In image coding the image is compressed while conserving the information. Recently there have been attempts to improve the compression rate by recognizing the texture in an image and replacing it with a symbolic representation.

Computer graphics has another approach to texture. Textures are used to mimic natural scenes. The problem is reversed. How to generate a natural texture? The effort to synthesize textures has also had an effect to texture recognition. Some texture recognition methods based on texture models have been developed.

There is also research going on to extract three-dimensional information from two dimensional images. This extraction is based on determining surface orientations. Surfaces are often textured. The surface orientation is determined by identifying the texture and the texture projection .

4.1. Texture Definitions

Even though texture is an intuitive concept, a formal definition of texture has proven elusive in the literature[38,20,21,1,17,35] over the years. In 1973, Haralick, Shanmugam and Dinstein noted, “texture has been extremely refractory to precise definition.” Many researchers have expressed this sentiment. Cross and Jain, “There is no universally accepted definition for texture.” Bovik, Clarke and Geisler, “an exact definition of texture either as a surface property or as an image property has never been adequately formulated.” And Jain and Karu, “Texture eludes a formal definition.”

Despite this lack of a universally agreed definition, all researchers agree on two points. Firstly, there is significant variation in intensity levels between nearby pixels; that is, at the limit of resolution, there is non-homogeneity. Secondly, texture is a homogeneous property at some spatial scale larger than the resolution of the image. Haralick summarises these two points by his definition, "A fundamental characteristic of texture is it cannot be analysed without a frame of reference of tonal primitive being stated or implied. For any smooth grey 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."

Some researchers describe texture in terms of the human visual system. They claim textures do not have uniform intensity, but are none-the-less perceived as homogeneous regions by a human observer. For example, Bovik, Clarke and Geisler write, “an image texture may be defined as a local arrangement of image irradiances projected from a surface patch of perceptually homogeneous irradiances.” Also, Chaudhuri, Sarkar and Kundu write, “Texture regions give different interpretations at different distances and at different degrees of visual attention. At a standard distance with normal attention, it gives the notion of macro-regularity that is characteristic of the particular texture. When viewed closely and attentively, homogeneous regions and edges, sometimes constituting texels are noticeable.” However, a definition based on human acuity poses problems when used as the theoretical basis for a quantitative texture analysis algorithm. Faugeras and Pratt note, “The basic pattern and repetition frequency of a texture sample could be perceptually invisible, although quantitatively present.”

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. A complete definition for texture is really challenging, nevertheless many experts successed it over the years. Haralick gives the following definition to the texture, " Image texture is described by the number and types of its primitives and the spatial organization or layout of its primitives. The spatial organization may be random, may have a pairwise dependence of one primitive on a neighbouring primitive, or may have a dependence of n primitives at a time. The dependence may be structural, probabilistic, or functional." And the definition given by Van Gool, "Texture can be defined as a structure composed of many of more or less ordered similar elements or patterns without one of these drawing special attention. So a global unitary impression is offered to the observer. We could think of a strictly ordered array of identical subpatterns like a chessboard for instance. Such a texture is called deterministic. It can be described by the characteristics of one such subpattern or primitive and by the placement rules defining the spatial distribution of the primitives. We could also have in mind a pattern merely obeying some statistical laws. The resulting structure might resemble noise on a television screen. Such a texture is said to be stochastic. We have to point out, however, that deterministic textures can be heavily disturbed in their repetitiousness and their primitives might be similar, but not identical at all. Then the texture is no longer an ideal one but is referred to as an observable texture.. In fact it is in between the deterministic and the stochastic model."

Identifying the perceived qualities in an image is an important first step towards building mathematical models for texture. The intensity variations in an image which characterize texture are generally due to some underlying physical variation in the scene such as pebbles on a beach or waves in water. Modelling this physical variation is very difficult, so texture is usually characterized by the two-dimensional variations in the intensities present in the image. This explains the fact that no precise, general definition of texture exists in the computer vision literature.

What we can claim with confidence is that texture is a contextual property of gray level values in a spatial neighborhood whose size depends upon the texture type. So it is a property of pixel areas. In the next section, we’ll see the categories of texture.

4.2.Texture Taxanomy

Due to its wide variability, people usually describe[2,3,37,38,57,1,17] texture as fine, coarse, grained, smooth, etc., implying that some more precise features must be defined to make machine recognition possible. Such features can be found in the tone and structure of a texture. Tone is mostly based on pixel intensity properties in the primitive, while structure is the spatial relationship between primitives. A texture primitive, a texel, is a contiguous set of pixels with some tonal and/or regional property, and can be described by its average intensity, maximum or minimum intensity, size, shape, etc.. The texels may have various sizes and degrees of uniformity, may be oriented in various directions, may be spaced at varying distances in different directions and may have various magnitudes and variations of contrast and opacity. The placement of a texel in texture could be periodic, quasi-periodic or random. Natural textures have generally random texels, whereas artificial textures have deterministic or periodic ones. Let's see an example:

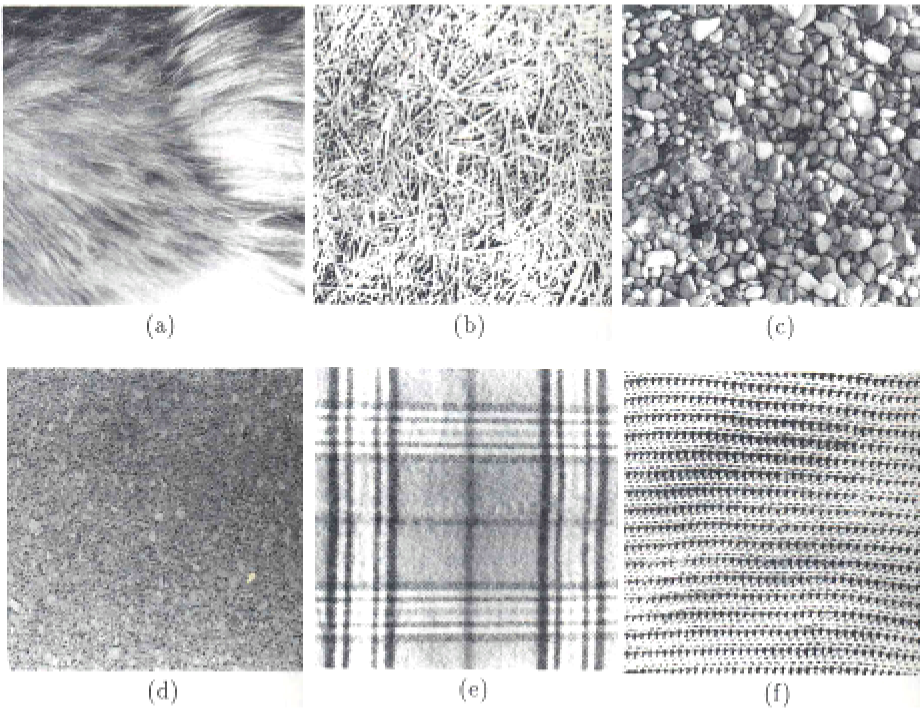
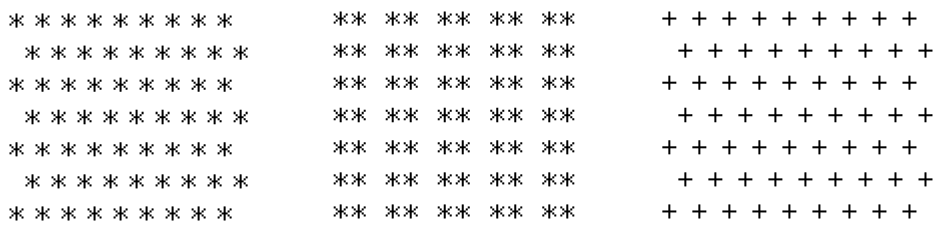


Figure 4.1: Texture examples

(a) dog fur (b) grass (c) river pebbles (d) cork (e) chequered textile (f) knitted fabric

Primitives in grass and dog fur are represented by several pixels and correspond to a stalk or a pile; cork is built from primitives comparable in size with pixels. It is difficult, however, to define primitives for the chequered textile or fabric which can be defined in at least two hierarchical levels. The first level of primitives corresponds to textile checks or knitted stripes, and the second the finer texture of the fabric or individual stitches.

Image texture as a global representation is described by the number and types of primitives and by their spatial relationship but it doesn’t mean that we have optimal differentiability for all textures. Figures a and b show that the same number and the same type of primitives does not necessarily give the same texture.



(a) (b) (c)

Figure 4.2: Artificial textures

Similarly, Figures 4.2a and 4.2c show that the same spatial relationship of primitives does not guarantee texture uniqueness, and therefore is not sufficient for texture description.

Texture tone and structure are not independent; textures always display both tone and structure even though one or the other usually dominates, and we usually speak about one or the other only. Tone can be understood as tonal properties of primitives, taking primitive spatial relationships into consideration. Structure refers to spatial relationships of primitives considering their tonal properties as well. If the texture primitives in the image are small and if the tonal differences between neighbouring primitives are large, a fine texture results, e.g. d in the Figure 4.1. If the texture primitives are larger and consist of several pixels, a coarse texture results, e.g. c in the Figure 4.1. Note that the fine/coarse texture characteristic depends on scale. Further, textures can be classified according to their strength, which then influences also the choice of texture description method. Weak textures have small spatial interactions between primitives, and can be adequately described by frequencies of primitive types appearing in some neighbourhood. Because of this, many statistical texture properties are evaluated in the description of weak textures. In strong textures, the spatial interactions between primitives are somewhat regular. To describe strong textures, the frequency of occurrence of primitive pairs in some spatial relationship may be sufficient. Strong texture recognition is usually accompanied by an exact definition of texture primitives and their spatial relationships.

It remains to define a constant texture. An image region has a constant texture if a set of its local properties in that region is constant, slowly changing, or approximately periodic. The set of local properties can be understood as primitive types and their spatial relationships. An important part of the definition is that the properties must be repeated inside the constant texture area. How many times must the properties be repeated? Assume that a large area of constant texture is available, and consider smaller and smaller parts of that texture, digitizing it at constant resolution as long as the texture character remains unchanged. Alternatively, consider larger and larger parts of the texture, digitizing it at constant raster, until details become blurred and the primitives finally disappear. We see that image resolution is a consistent part of the texture description; if the image resolution is appropriate, the texture character does not change for any position in our window, we obtain constant textures.

Most texture research can he characterized by the underlying assumptions made about the texture formation process described above. Two main texture description approaches exist: statistical (stochastic) and structural (syntactic). The choice of the assumption depends primarily on the type of textures.

Statistical methods yield characterizations of textures as smooth, coarse, grainy, etc. and are suitable if texture primitive sizes are comparable with the pixel sizes. Textures that are random in nature like sand, water, grass are well suited for statistical characterization. The structural placement paradigm for textures may also include a random aspect; in the stochastic point of view, however, we take a more extreme position and consider that the texture is a sample from a probability distribution on the image space like noise on a television screen. The image space is usually an N X N grid and the value at each grid point is a random variable in the range {0,1,…,G-1}.

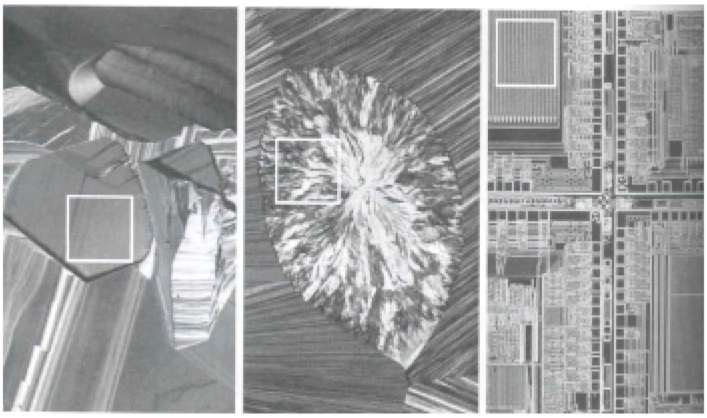
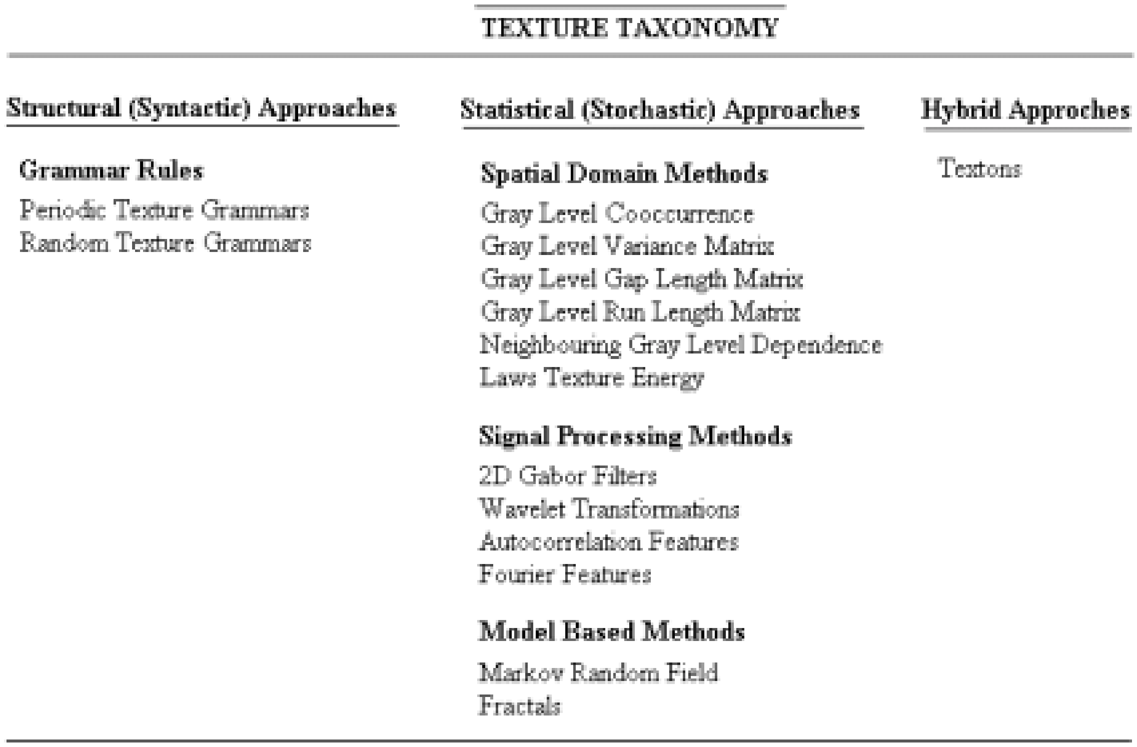


Figure 4.3: Smooth, coarse, and regular textures

Syntactic (structural) methods are more suitable for textures where primitives can be assigned a label, meaning that primitive type can be described using a larger variety of properties like shape, size than just tonal properties. Purely syntactic texture description models are based on the idea that textures consist of primitives located in almost regular relationships. As an example, we could think of a strictly ordered array of identical sub-patterns like a checkerboard. The sub-patterns may be of deterministic shape, such as circles, hexagons, or even dot patterns. Macrotextures have large primitives, whereas microtextures are composed of small primitives. These terms are again relative to the image resolution.

On the other hand, textures of the real world are usually irregular in which structural errors, distortions, or even structural variations are frequent. This means that no strict rule can be used to describe a texture in reality. To make syntactic description of real textures possible, variable rules must be incorporated. Nevertheless, purely syntactic methods of texture description are accompanied by many difficulties with syntactic analyser learning and with complex grammar inference. This is the main reason why purely syntactic methods are not widely used. On the other hand, a precise definition of primitives brings many advantages and it is not wise to avoid it completely. Hybrid methods of texture description combine the statistical and syntactic approach. The technique proposes deterministic primitives and probabilistic relations between primitives.

The broad taxonomy map below is derived by combining various charts from the literature, still doesn’t claim to be an absolute truth due to the variability of the texture definitions; In the thesis, we didn't include the results of the syntactic, fractals and wavelet based methods. All the other algorithms are studied and implemented as discussed on the following two chapters.



TEXTURE TAXONOMY

Structural (Syntactic) Approaches	Statistical (Stochastic) Approaches	Hybrid Approches
Grammar Rules	**Spatial Domain Methods**	Textons
Periodic Texture Grammars	Gray Level Cooccurrence	
Random Texture Grammars	Gray Level Variance Matrix	
	Gray Level Gap Length Matrix	
	Gray Level Run Length Matrix	
	Neighbouring Gray Level Dependence	
	Laws Texture Energy	
	Signal Processing Methods	
	2D Gabor Filters	
	Wavelet Transformations	
	Autocorrelation Features	
	Fourier Features	
	Model Based Methods	
	Markov Random Field	
	Fractals	

Figure 4.4: Texture taxonomy

4.3. Texture Algorithms

4.3.1. Statistical Approaches

Statistical approaches[37,57,20,11] define the texture as “a quantitative measure of the arrangement of intensities in a region”. Each texture is described by a feature vector of properties, which represents a point in a multi-dimensional feature space. The statistical approaches include spatial domain methods, signal processing methods and model based methods which algorithms will be explained in the following sections.

In spatial domain methods, features are derived from especially second order statistics of the texture gray level values because human beings are sensitive to second-order statistics. Examples of such statistics are the gray level cooccurrence matrix and the gray level variance matrix. Higher order statistics such as the gray level run length, the Fourier power spectrum and the autocorrelation function can also be measured.

Frequency based methods like Gabor filters and Wavelet transformations take the visual system as its model and decomposes images into a number of band-pass filtered images, each of which contains intensity variations over a narrow range of frequency (size) and orientation, which will resemble certain textural properties. Psychophysical experiments have shown that methods based on spatial-frequency domain features are plausible with the human visual system. In recent years, by the development of the wavelet theory this method has received special interest, number of algorithms based on wavelet transforms and Gabor filters have been proposed for classification and segmentation of textured images. But for this thesis study, wavelet based solutions are excluded. The other frequency features of the texture like Fourier transformations, autocorrelation functions can be included under the signal processing subgroup; such methods will be studied in the following sections.

Model based texture analysis methods try to capture the process that generated the texture by determining the parameters of a predefined model. Model based methods assume textured images as realizations or samples from parametric probability distributions on the image space, and try to fit simultaneous autoregressive models, Markov random field models, and fractal models to the textured image. Model parameters are used as features in the classification and segmentation problems. Advantage of this method over other statistical methods is that the model parameters can be used not only to describe the texture but also to synthesize it. They have been used for classification, segmentation and compression of textured images.

4.3.1.1. Histogram Statistics

One of the simplest approaches[4,5,38] for describing texture is to use statistical moments of the gray-level histogram of an image or region. Let z be a random variable denoting gray levels and let $p(z_i)$, i = 0,1,2,..., L-1, be the corresponding histogram, where L is the number of distinct gray levels. The nth moment of z about the mean is

$$\mu_n(z) = \sum_{i=0}^{L-1} (z_i - m)^n p(z_i) \tag{4.1}$$

where m is the mean value of z, the average gray level:

$$m = \sum_{i=0}^{L-1} z_i p(z_i) \tag{4.2}$$

Note from equation 4.1 and 4.2 that $\mu_0=1$ and $\mu_1=0$. The second moment, the variance $\sigma^2(z)= \mu_2(z)$, is of particular importance in texture description. It is a measure of gray-level contrast that can be used to establish descriptors of relative smoothness. For example, the measure

$$R = 1 - \frac{1}{1+\sigma^2(z)} \tag{4.3}$$

is 0 for areas of constant intensity, the variance is zero there, and approaches 1 for large values of $\sigma^2(z)$. Because variance values tend to be large for greyscale images, it is a good idea to normalize the variance to the interval [0 1]. This is done simply by dividing $\sigma^2(z)$ by $(L-1)^2$. The standard deviation, $\sigma(z)$, is also used frequently as a measure of texture because values of standard deviation tend to be more intuitive to many people.

The third moment,

$$\mu_3(z) = \sum_{i=0}^{L-1} (z_i - m)^3 p(z_i), \tag{4.4}$$

is a measure of the skewness of the histogram while the fourth moment is a measure of its relative flatness. The fifth and higher moments are not so easily related to histogram, but they do provide further quantitative discrimination of texture content.

Some useful additional texture measures based on histograms include a measure of ‘uniformity’, given by

$$U = \sum_{i=0}^{L-1} p^2(z_i), \tag{4.5}$$

and an average entropy measure is defined as,

$$e = -\sum_{i=0}^{L-1} p(z_i)\log_2 p(z_i) \tag{4.6}$$

Because the p’s have values in the range [0,1] and their sum equals to 1, the measure U is maximum for an image in which all gray levels are equal, maximally uniform. Entropy is a measure of variability, 0 for a constant image.

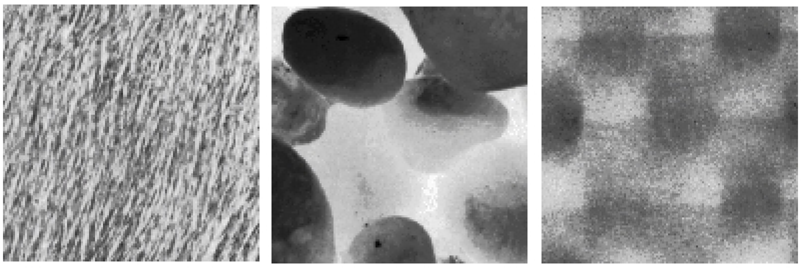


Figure 4.5: Texture examples from Brodatz database

Texture	Mean	Standard Deviation	R (normalized)	Third Moment	Uniformity	Entropy
Smooth	60.4	9.8	0.01	-0.01	0.035	4.7
Coarse	136.7	67.9	0.07	0.10	0.010	8.1
Regular	85.5	40.2	0.03	0.22	0.027	5.9

Table 4.1: Histogram statistics

Table 4.1 summarizes the values of the preceding measures for the three types of textures highlighted in Figure 4.5. The mean just tells us the average gray level of each region and is useful only as a rough idea of intensity, not really texture. The standard deviation is much more informative; the numbers clearly show that the first texture has significantly less variability in gray level it is smoother than the other two textures. The coarse texture shows up clearly in this measure. The same comments hold for R, because it measures essentially the same thing as the standard deviation. The third moment generally is useful for determining the degree of symmetry of histograms and whether they are skewed to the left, negative value, or the right, positive value. This gives a rough idea of whether the gray levels are biased toward the dark or light side of the mean. In terms of texture, the information derived from the third moment is useful only when variations between measurements are large. Looking at the measure of uniformity, we again conclude that the first sub-image is smoother, more uniform than the rest and that the most random, lowest uniformity corresponds to the coarse texture. Finally, the entropy values are in the opposite order and thus lead us to the same conclusions as the uniformity measure did. The first sub-image has the lowest variation in gray level and the coarse image the most. The regular texture is in between the two extremes with respect to both of these measures.

There is one more way to work with moment features of a texture, spatial moments:

The $(p+q)^{th}$ moments over an image region *R* are given by the formula

$$m_{pq} = \sum_{(x,y)\in R} x^p y^q I(x,y) \tag{4.7}$$

where *I(x,y)* is the image matrix, a two dimensional integer array. If the region *R* is a local rectangular area and the moments are computed around each pixel in the image, then this is equivalent to filtering the image by a set of spatial masks. The resulting filtered images that correspond to the moments are then used as texture features. The masks are obtained by defining a window of size *W*x*W* and a local coordinate system centred within the window. Let (i, j) be the image coordinates at which the moments are computed. For pixel coordinates (*m, n*) which fall within the *W*x*W* window centred at (*i, j*), the normalization coordinates (x_m, y_n) are given by the formula:

$$x_m = \frac{(m-i)}{(W/2)} \qquad y_m = \frac{(n-j)}{(W/2)} \tag{4.8}$$

Then, the moments within a window centred at pixel (*i, j*) are computed by the sum in equation 4.7 that uses the normalized coordinates:

$$m_{pq} = \sum_{n=-W/2}^{W/2} \sum_{m=-W/2}^{W/2} x_m^p y_n^q I(m,n) \tag{4.9}$$

The coefficients for each pixel within the window to evaluate the sum are what define the mask coefficients. If *R* is a 3x3 region, then the resulting masks are given in Figure 4.6. The size *W* of the window can be adjusted if 3x3 windows can not define the textural properties adequately. Especially for textures with large low frequency components i.e. slowly varying textures windows of larger sizes may be more appropriate.

$$M_{00}=\begin{array}{|c|c|c|}\hline 1 & 1 & 1 \\ \hline 1 & 1 & 1 \\ \hline 1 & 1 & 1 \\ \hline\end{array}\qquad M_{10}=\begin{array}{|c|c|c|}\hline -1 & -1 & -1 \\ \hline 0 & 0 & 0 \\ \hline 1 & 1 & 1 \\ \hline\end{array}\qquad M_{01}=\begin{array}{|c|c|c|}\hline -1 & 0 & 1 \\ \hline -1 & 0 & 1 \\ \hline -1 & 0 & 1 \\ \hline\end{array}$$

$$M_{20}=\begin{array}{|c|c|c|}\hline 1 & 1 & 1 \\ \hline 0 & 0 & 0 \\ \hline 1 & 1 & 1 \\ \hline\end{array}\qquad M_{11}=\begin{array}{|c|c|c|}\hline 1 & 0 & -1 \\ \hline 0 & 0 & 0 \\ \hline -1 & 0 & 1 \\ \hline\end{array}\qquad M_{02}=\begin{array}{|c|c|c|}\hline 1 & 0 & 1 \\ \hline 1 & 0 & 1 \\ \hline 1 & 0 & 1 \\ \hline\end{array}$$

Figure 4.6: Mask coefficients corresponding to first and second moments

Measures of texture computed using only histograms suffer from the limitation that they carry no information regarding the relative position of pixels with respect to each other. One way to bring this type of information into the texture-analysis process is to consider not only the distribution of intensities, but also the positions of pixels with equal or nearly equal intensity values. The cooccurence probabilities are derived for this reason, which will be discussed in detail after the autocorrelation function approach.

4.3.1.2. Autocorrelation (ACF)

Coarse textures are built from larger size primitives, fine textures from smaller primitives. Fine textures are characterized by higher spatial frequencies, coarse textures by lower spatial frequencies.

Autocorrelation shows[3,37,38,4,11,1,16] up both local intensity variations and also the repeatability of the texture. In an autocorrelation model, a single pixel is considered a texture primitive and primitive tone property is the gray level. If the texture primitives are relatively large, the autocorrelation function value decreases slowly with increasing distance, while it decreases rapidly if texture consists of small primitives. If primitives are placed periodically in a texture, the autocorrelation increases and decreases periodically with distance.

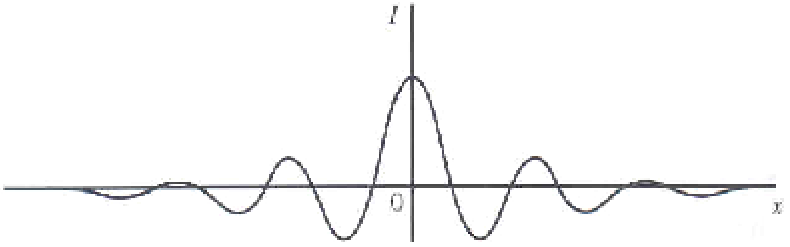


Figure 4.7: 1-D profile of the autocorrelation function

The Figure 4.7 shows the possible 1-D profile of the ACF for a piece of material in which the weave is subject to significant spatial variation; notice that the periodicity of the autocorrelation function is damped down over quite a short distance.

Mathematically, the spatial size of tonal primitives, i.e. texels, in texture can be represented by the width of the spatial ACF which is defined as:

$$r(k,l)=m_2(k,l)/m_2(0,0) \tag{4.10}$$

$$m_i(k,l)=\frac{1}{N_w}\sum_{(m,n)\in W}\sum[f(m-k,n-l)]^i \tag{4.11}$$

where k,l is the position difference in the m,n direction, f(m,n) is the image, two dimensional integer array, i=1,2,.. and N_w is the number pixels over a small moving window W.

The coarseness of texture is expected to be proportional to the width of the ACF which can be represented by distances x_0, y_0 such that

$$r(x_0,0)=r(0,y_0)=1/2. \tag{4.12}$$

Other measures of spread of the ACF are obtained via the moment-generating function:

$$M(k,l)\underline{\Delta}\sum_m\sum_n(m-\mu_1)^k(n-\mu_2)^l r(m,n) \tag{4.13}$$

where

$$\mu_1\underline{\Delta}\sum_m\sum_n mr(m,n), \qquad \mu_2\underline{\Delta}\sum_m\sum_n nr(m,n) \tag{4.14}$$

Features of special interest are the profile spreads M(2,0) and M(0,2), the cross-relation M(1,1), and the second-degree spread M(2,2).

The calibration of the ACF spread on a fine-coarse texture scale depends on the resolution of the image. This is because a seemingly flat region, no texture, at a given resolution could appear as fine texture at higher resolution and coarse texture at lower resolution.

The ACF by itself is not sufficient to distinguish among several texture fields because many different image ensembles can have the same ACF.

4.3.1.3. Cooccurrence Matrices and Features

The gray level cooccurrence matrix approach is based on studies of the statistics of pixel intensity distributions[2,3,4,5,20,37,38,57,21,1,16,14,31,36,45]. As hinted above in the histogram section, with regard to the variance in pixel intensity values, single pixel statistics do not provide rich enough descriptions of textures for practical applications. Thus, it is natural to consider second-order statistics obtained by considering pairs of pixels in certain spatial relations to each other. Co-occurrence matrices express the relative frequencies or probabilities C(i, j | d, θ) with which two pixels having relative polar coordinates (d, θ) appear with intensities i, j.

A co-occurrence matrix is a two-dimensional array C in which both the rows and the columns represent a set of possible pixel values V. For example, for gray-tone images V can be the set of possible gray tones and for colour images V can be the set of possible colours. The value of C(i, j) indicates how many times value i cooccurs with value j in some designated spatial relationship. For example, the spatial relationship might be that value i occurs immediately to the right of value j. To be more precise, we will look specifically at the case where the set V is a set of gray tones and the spatial relationship is given by a vector d that specifies the displacement between the pixel having value i and the pixel having value j. Let d be a displacement vector (dr, dc) where dr is a displacement in rows (downward) and dc is a displacement in columns (to the right). This format is good to represent both the direction and displacement. Let V be a set of gray tones. The gray-tone co-occurrence matrix C_d for image I is defined by

$$C_d[i,j] = \left|\left\{[r,c] \quad | \quad I[r,c] = i \quad and \quad I[r+dr, c+dc] = j\right\}\right| \tag{4.15}$$

Figure 4.8 below illustrates this concept with a 4 x 4 image I and three different cooccurrence matrices for I. If the image has *G* gray levels, then the density functions can be written as *GxG* matrices. Each matrix can be computed from a digital image by counting the number of times each pair of gray levels occurs at separation *d* and in the direction specified by *Θ*.

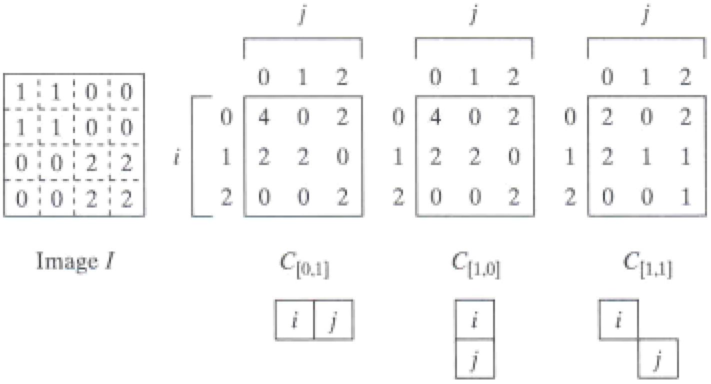


Figure 4.8: Three different cooccurrence matrices for a greyscale image

In $C_{[0,1]}$ note that position [1,0] has a value of 2, indicating that j=0 appears directly to the right of i=1 two times in the image. However, position [0,1] has a value of 0, indicating that j=1 never appears directly to the right of i = 0 in the image. The largest cooccurence value of 4 is in position [0,0], indicating that a 0 appears directly to the right of another 0 four times in the image.

For this thesis, it is assumed that the textural information is adequately specified by the full set of four ($\Theta = 0^{\circ}$, $\Theta = 45^{\circ}$, $\Theta = 90^{\circ}$, $\Theta = 135^{\circ}$) spatial gray level dependence matrices.

There are two important variations of the standard gray-tone co-occurrence matrix. The first is the normalized gray-tone co-occurrence matrix N_d defined by

$$N_d[i,j] = \frac{C_d[i,j]}{\sum_i \sum_j C_d[i,j]} \qquad (4.16)$$

which normalises the cooccurrence values to lie between zero and one and allows them to be thought of as probabilities in a large matrix.

The second is the symmetric gray-tone co-occurrence matrix S_d defined by

$$S_d[i,j] = C_d[i,j] + C_{-d}[i,j] \qquad (4.17)$$

which groups pairs of symmetric adjacencies. C_{-d} is just the transpose C_d .

Cooccurence matrices capture properties of a texture, but they are not directly useful for further analysis, such as comparing two textures. These data must be condensed to relatively few numbers before they can be used to classify the texture. The early paper by Haralick et al. in 1973 gave 14 such measures; However, Conners and Harlow in 1980 found that only five of these measures were normally used, namely "energy", "entropy", "correlation", "local homogeneity" and "inertia". The following are concurrent standard features derivable from a normalized co-occurrence matrix. Texture classification can be based on criteria derived from these features.

Uniformity of energy or angular second moment: It is an image homogeneity measure-the more homogeneous the image, the larger the value.

$$\mathit{Energy} = \sum_i \sum_j N_d^2[i, j] \tag{4.18}$$

Entropy: Entropy gives a measure of complexity of the image. Complex textures tend to have higher entropy.

$$\mathit{Entropy} = -\sum_i \sum_j N_d[i, j] \log_2 N_d[i, j] \tag{4.19}$$

Contrast: A measure of local image variations present and image contrast

$$\mathit{Contrast} = \sum_i \sum_j (i - j)^2 N_d[i, j] \tag{4.20}$$

Homogeneity:

$$\mathit{Homogeneity} = \sum_i \sum_j \frac{N_d[i, j]}{1 + |i - j|} \tag{4.21}$$

Correlation: It is a measure of image linearity. Linear directional structures in direction ϕ result in large correlation values in this direction.

$$\mathit{Correlation} = \frac{\sum_i \sum_j (i - \mu_i)(j - \mu_j) N_d[i, j]}{\sigma_i \sigma_j} \tag{4.22}$$

where μ_i , μ_j are the means and σ_i, σ_j are the standard deviations of the row and column sums $N_d[i]$ and $N_d[j]$ defined by

$$N_d[i] = \sum_j N_d[i, j] \tag{4.23}$$

$$N_d[j] = \sum_i N_d[i, j] \tag{4.24}$$

where μ_x, μ_y are means and σ_x, σ_y standard deviations

After one extracts these features using the matrices evaluated for four different directions and various *d* values, he is ready to use them to identify the texture. The basic assumption is that the features extracted from matrices that are formed using a few number of *d* values and four directions contain sufficient textural information to identify the texture. One problem with deriving texture measures from co-occurrence matrices is how to choose the displacement vector d. A solution suggested by Zucker and Terzopoulos is to use a χ^2 statistical test to select the value(s) of d that have the most structure; that is, to maximize the value:

$$\chi^2(d) = \left(\sum_i \sum_j \frac{N_d^2[i,j]}{N_d[i]N_d[j]} - 1 \right) \tag{4.25}$$

The co-occurrence method describes second-order image statistics and works well for a large variety of textures. Good properties of the co-occurrence method are the description of spatial relationship between tonal pixels, and invariance to monotonic gray level transformations. If an appropriate quantization can be applied to the image, a fast defect inspection can be done. Depending on the types of the textures considered, the matrices for some directions and/or some *d* values might not be evaluated thus time may be saved.

On the other hand, it does not consider primitive shapes, and therefore cannot be recommended if the texture consists of large primitives. The co-occurrence approach is not suited to work with textures composed of large patches. As stated earlier, most of the textural information is assumed to be obtained by using certain number of matrices. However, this might not be the case especially for complicated textures. Although the choice of four directions might be adequate in the directional view, the choice of *d* values plays a very important role. While some textures can be identified by small *d* values, others might be discriminated only if large *d* values are used. The selection of the suitable *d* values is totally texture-dependent and this decreases the robustness of the algorithm. For eight bit gray level images the size of each co-occurrence matrix is 256x256 which brings computation and memory problems. The number of gray levels may be set to 32 or 64 which decreases the co-occurrence matrix sizes, but loss of gray level accuracy is a resulting negative effect although this loss is usually insignificant in practice.

4.3.1.4. Gray Level Variance Matrix(GLVM)

Gray level variance matrices[64,20] give the gray level variance around a pixel within a window of size WxW. The matrix element $p(i, j|w)$ contains the total number of occurrences of gray level with variance i within a window of size WxW centred on pixel with gray level j. Several texture features can be extracted from this matrix. Changing the size of the windows introduces new sets of feature values.

Some of the features that can be evaluated using gray level variance matrices are listed below:

Smoothness (SM) $$\sum_{j=1}^{G}\sum_{i=1}^{v} p(i,j)/N \qquad (4.26)$$

where G is number of gray levels in the image. The total number of pixels within the region of interest, N, is used as a normalizing factor. The value of v (> 0) has to be chosen as small as possible such that a window with variance ≤ $(v-1)$ can be called a homogeneous window or region. For example if $v = 1$ then all the windows with variance zero (v-1) or the windows having pixels with same gray level will be considered. SM will get high values for images which have many smooth windows of size WxW.

Gray Level Variance Ratio (GLVR) $$\frac{\sum_{j=\mu}^{G}\sum_{i=1}^{V} p(i,j)}{\sum_{j=1}^{\mu-1}\sum_{i=1}^{V} p(i,j)} \qquad (4.27)$$

where V is the highest variance and μ is the mean gray value. Images which have lightly condensed particles with small local variation will give high GLVR values.

Minimum Gray Level (MGL) $MIN(j) \qquad p(i,j) \neq 0$ (4.28)

Images with dark areas will give low MGL values.

Roughness (RH) $$[MAX\ (i) - MIN\,(i)] \\ \text{x}[MAX\ (j) - MIN\,(j)] \quad p(i,j) \neq 0 \qquad (4.29)$$

Images with high local variation and large gray level range will give high RH values.

Low Variance Emphasis (LVE) $$\sum_{j=1}^{G}\sum_{i=1}^{V} \frac{p(i,j)/N}{j^2} \qquad (4.30)$$

Images, which have homogeneous areas, areas with low variance, will give high LVE values.

High Variance Emphasis (HVE) $\sum_{j=1}^{G}\sum_{i=1}^{V} j^2 \text{ x } p(i,j)/N$ (4.31)

HVE will be high for images with high local variance within a window.

Low Gray Level Variance Emphasis (LGVE) $\sum_{j=1}^{G}\sum_{i=1}^{V} \frac{p(i,j)/N}{i^2}$ (4.32)

High Gray Level Variance Emphasis (HGVE) $\sum_{j=1}^{G}\sum_{i=1}^{V} i^2 \text{ x } p(i,j)/N$ (4.33)

These features may be evaluated for different window sizes depending on the nature of the texture that is being examined. Local relationships between pixels affect the feature values.

Besides the advantages that co-occurrence matrices have, gray level variance matrices extract local information of a texture image which may be useful to synthesize that particular texture.

On the other hand, like gray level co-occurrence matrices, gray level variance matrices suffer from high computational cost.

4.3.1.5. Gray Level Gap Length Matrix

A gray level gap length matrix is defined[63,20] in order to obtain structural information for each gray level in the image. The gray level gap length (GLGL) method is based on measuring the distribution of gray level gap lengths for each gray level in an image. A gap for gray level g occurs when g is only found at the beginning and the end of a set of consecutive, collinear pixels, while all pixel values in between are either above or below g. The gap length is the distance between these two pixels minus one. Thus, two neighbouring pixels with identical gray level have zero gap length. In the case where we never meet another pixel with gray level g again, the gap length is considered as infinite and is omitted.

The gray level gap length matrix (GLGLM) is a *2-D* array, $A_n(g,l \backslash \Theta)$, where g is the gray level, l is the gap length, and Θ is the given direction. The number of rows is equal to the number of gray levels G in the image, and the number of columns is the maximum gap length, as obtained when searching along a direction Θ. This matrix can be seen as a complement to the gray level run length matrix (GLRLM). It gives the size distribution of texture elements for a given direction in the image.

Given an image of *MxN* pixels with *G* gray levels from 0 to *G*-1, let *f(i,j)* be the gray level function at pixel (*i,j*), and *L* be the maximum gap length. The element of GLGLM at angle *Θ* is defined as:

$$A(g,l/\Theta) = Card\begin{bmatrix} (i,j) \mid f(i,j) = f(i+x, j+y) = g, \\ f(i+u, j+v) \neq g \\ 0 < u < x, 0 < v < y \end{bmatrix} \tag{4.34}$$

$$\begin{gathered} x = (l+1)\cos\Theta,\ y = (l+1)\sin\Theta \\ 0 \le g \le G-1,\ 0 \le l \le L,\ 0^0 \le \Theta \le 180^0 \end{gathered} \tag{4.35}$$

where "Card" stands for "the number of".

For an image with isotropic texture patterns, an average matrix from GLGMs, obtained in different directions will give a rotation invariant representation.

As in the case of gray level cooccurrence matrices, the number of occurrences decreases as the gap length increases. For statistics at each gap length to be comparable, normalisation has to be done. The normalized GLGLM, $A_n(g,l\backslash\Theta)$ can be obtained by:

$$A_n(g,l\backslash\Theta) = \frac{A(g,l\backslash\Theta)}{W(l)} \tag{4.36}$$

where the weighting function *W(l)* varies according to gap length *l*. For the analysis along the *x* or *y* direction only, the weighting function can be chosen as follows:

when $l \le G$

$$W(l) = (M-x)(N-y) \tag{4.37}$$

when $l > G$

$$W(l) = W_1(x)W_2(y) \tag{4.38}$$

where

$$W_1(x) = \begin{Bmatrix} (\frac{M}{x} - 1)(G-1) + 2 & x \neq 0 \\ M & x = 0 \end{Bmatrix} \tag{4.39}$$

$$W_2(y) = \begin{Bmatrix} (\frac{N}{y} - 1)(G-1) + 2 & y \neq 0 \\ N & y = 0 \end{Bmatrix} \tag{4.40}$$

The special consideration for the case where $l > G$ is due to the fact that no repetition of gray level *g* is allowed in a gap length *l*. So, the weighting function is approximately the maximum number of gaps of a length *l* in an image.

Normally, only a few gray levels are used in the GLGL matrix to save computational time and storage, as well as to increase the possibility of finding significant gaps. Some gray levels are considered more significant than others because they form the high contrast structures of the texture. Histogram transformation may be performed first, in order to extend the dynamic range of gray levels. Then the image is re-quantified into a reduced number of gray levels. Since adjacent gray levels normally catch similar structural information, a sum of gap lengths among adjacent gray levels may be able to find the size distribution of the structure, especially in the case of uneven illumination. For example, the size distributions for darker gray levels and brighter gray levels can be obtained by:

$$S_1(l) = \sum_{g=0}^{t} A_n(g,l), \qquad S_2(l) = \sum_{g=t+1}^{G-1} A_n(g,l) \tag{4.41}$$

where $0 \leq l \leq L$, and L is limited by the size of the image. Here, *t* is some chosen threshold, e.g. $t = G/2\text{-}1$. The *S* values calculated can be used to estimate the period of repetitive patterns in a texture. When one obtains the graphs of S_1 and S_2, the highest peaks for dark and bright gray levels indicate the estimated quasi-periodicities. The period is just the sum of these two quasi-periodicities.

Texture statistics are normally classified into different orders according to the context information involved. First-order statistics extract parameters from individual pixels. Second-order statistics describe the joint gray level distribution pairs of pixels, given a geometrical relation. In order to get a complete structural description, a series of second order cooccurrence matrices at different distances are to be calculated. Higher order statistics describe the relationship of three or more pixels, and mix the global information together into one data set. Therefore, they are suitable for finding structural characteristics in an image; Gray level gap length matrix reflects directly the size distribution of texture elements, and the calculation of the matrix and its features are both simple and fast. After the image is re-quantified the GLGL algorithm works fast. If the quantization may be done by the aid of a look-up table defined in the hardware, very fast implementations are possible. For periodicity detection, features extracted from the normalized GLGL works much faster than the commonly used gray level cooccurrence matrices (GLCM), and provides very fast and memory-efficient granulometry. The size features, are very useful in unsupervised texture segmentation and classification.

On the other hand, for saving computational time and storage, a pre-processing of the image is required to quantize the image into a reduced number of gray levels, not to lose more time.

4.3.1.6. Gray Level Run Length (Primitive length) Matrices

A large number of neighbouring pixels of the same gray level represents a coarse texture, a small number of these pixels represents a fine texture and the lengths of texture primitives in different directions can serve as a texture description. A primitive is a maximum contiguous set of constant gray level pixels located in a line these can then be described by gray level, length, and direction. The texture description features can be based on computation of continuous probabilities of the length and the gray level of primitives in the texture.

The gray level run length method[3,20,37] is based on computing the number of gray level runs of various lengths. A gray level run is a set of linearly adjacent picture points having the same gray level value. The length of the run is the number of picture points within the run. The element B(a, r | Ø) of the gray level run length matrix specifies the number of times a picture contains a run of length j for gray level a in the angle Ø direction. These matrices usually are calculated for several values of Ø.

Let B(a,r) be the number of primitives of the length r and gray level a, in some direction Θ; M, N the image dimensions, and L the number of image gray levels. Let N_r be the maximum primitive length in the image. The texture description features can be determined as follows; let K be the total number of runs

$$K = \sum_{a=1}^{L} \sum_{r=1}^{N_r} B(a,r) \tag{4.42}$$

The run length features proposed by Siew et. al. are defined as follows:

Long Primitives(Run) Emphasis (LRE) $$\frac{1}{K} \sum_{a=1}^{L} \sum_{r=1}^{N_r} B(a,r)r^2 \tag{4.43}$$

gives greater weight to long runs of any gray level.

Short Primitives (Run) Emphasis (SRE) $$\frac{1}{K} \sum_{a=1}^{L} \sum_{r=1}^{N_r} \frac{B(a,r)}{r^2} \tag{4.44}$$

gives greater weight to short runs of any gray level.

Gray Level Uniformity (GLU) $$\frac{1}{K}\sum_{a=1}^{L}\left(\sum_{r=1}^{N_r}B(a,r)\right)^2 \qquad (4.45)$$

This is smallest when runs are evenly distributed.

Primitive (Run) Length Uniformity (RLU) $$\frac{1}{K}\sum_{r=1}^{N_r}\left(\sum_{a=1}^{L}B(a,r)\right)^2 \qquad (4.46)$$

This is smallest when run lengths are evenly distributed.

Primitive (Run) Percentage (RPC) $$\frac{K}{\sum_{n=1}^{L}\sum_{r=1}^{N_r}rB(a,r)}=\frac{K}{MN} \qquad (4.47)$$

where N^2 is the number of points in the image. This is largest when the runs are all short.

The GLRLM is often used to extract statistical features which characterize the texture, since the longer runs imply homogeneity and shorter runs indicate rapid gray level changing.

On the other hand, the computational drawbacks like the other similar methods and the lack of gray level transition information are the drawbacks of GLRL matrices. From the viewpoint of one gray level, GLRLM is a partial measurement, it measures only the continuous parts. For an image with large and deep gaps, the neighbour gray levels will appear as many small runs, which may not give a proper description of the gaps.

4.3.1.7.Neighbouring Gray Level Dependence Matrices

This method[20] uses angular independent features, by considering the relationship between an element and all its neighbouring elements at one time instead of one direction at a time. This eliminates the angular dependency, at the same time reducing the calculation time required to process an image. It is based on the assumption that a gray level spatial dependence matrix (NGLDM) of an image can adequately specify the texture information. The matrix is computed from the gray level relationship between every element in the image and all its neighbours at a certain distance *d*. This matrix takes the form of a two-dimensional array, *Q*, where $Q(i, j)$ can be considered as frequency counts of greyness variation of an image. The size of the *Q* array is *GxNr* where *G* is the number of gray levels and *Nr* is the number of possible neighbours to a pixel in a range specified by *d*.

For an image function $f(i,j)$ it is easy to compute the Q matrix (for positive integer d, a) by counting the number of times the difference between each element in $f(i,j)$ and its neighbours is less than or equal a at a certain distance d. As an example, if the image is:

1	1	2	3	1
0	1	1	2	2
0	0	2	2	1
3	3	2	2	1
0	0	2	0	1

Figure 4.9: Example image

and d = 1, a = 0, then the NGLDM matrix is evaluated as follows: Each element corresponds to the total number of corresponding gray levels (determined by the row of the element) which have corresponding number of neighbours (determined by the column of the element) of the same gray level among the eight neighbours surrounding them.

	0	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**
0	0	0	1	0	0	0	0	0	0
1	0	0	1	1	0	0	0	0	0
2	0	0	0	0	4	1	0	0	0
3	0	1	0	0	0	0	0	0	0

Figure 4.10: NGLDM of the example image

This Q matrix relates to the texture information of an image. The degree of texture coarseness or fineness of an image is depicted by the distribution of entries in the Q matrix. Changing a and d can alter the distribution of the NGLDM numbers, because coarseness and fineness of an image is not absolute. Changing a and d also makes the NGLDM approach more effective for specific applications.

We obtain the numerical features from the Q matrix by using the functions where R is the normalizing factor [3]:

Small Number Emphasis (SNE) $\sum_{k}\sum_{s}[Q(k,s)/s^2]/R$ (4.48)

SNE is a measure of the fineness of the image. An image consisting of fine texture will have large values in the NGLDM columns making $Q(k,s)/s^2$ large for small s. Therefore the larger the SNE of an image is, the finer the image is.

Large Number Emphasis (LNE) $\quad \sum_k \sum_s [s^2 Q(k,s)]/R \quad$ (4.49)

LNE is a measure of the coarseness of an image. For a coarse texture picture, the large NGLDM numbers concentrate in the large s columns making $s^2 Q(k,s)$ larger for large s.

Number Non-uniformity (NNU) $\quad \sum_s \left[\sum_k [Q(k,s)] \right]^2 \Big/ R \quad$ (4.50)

Entropy (ENT) $\quad -\sum_k \sum_s Q(k,s)\log(Q(k,s))/R \quad$ (4.51)

NNU and ENT are related to the coarseness of an image.

Second Moment (SM) $\quad \sum_k \sum_s [Q(k,s)]^2 / R \quad$ (4.52)

The measurement of the homogeneity of the image is given by SM. Homogenous images will have large Q elements resulting in large SM values.

Since the Q matrix is essentially invariant under image rotation, any features extracted from the Q matrix are independent of angles. These are also invariant under linear gray level transformations or a linear shift of the gray level because the NGLDM numbers depend only on the relationship of the gray level values between an element and its neighbours.

The choice of the suitable a and d values is totally image dependent which forces one to change those parameters as the examined textures change.

4.3.1.8. Laws' Texture Energy Approach

The texture energy measures were developed[2,4,20,37,1] by Kenneth Ivan Laws in 1979 at the University of Southern California. This novel texture energy approach to texture analysis involved the application of simple filters to digital images. The basic filters he used were common Gaussian, edge detector and Laplacian filters, and were designed to highlight points of high "texture energy" in the image. By identifying these high energy points, smoothing the various filtered images, and pooling the information from them, he was able to characterize textures highly efficiently and in a manner compatible with pipelined hardware implementations. Laws' approach has strongly influenced much subsequent work.

Laws' approach to texture characterization consists of two main steps. First micro-statistic features are computed using 3x3, 5x5 or 7x7 convolution masks. Second, macro-statistic features are obtained over large windows. The most useful statistics are the sums of the squared or absolute values of the micro feature plane. The sum of squares justifies Laws' terminology “texture energy measure" but the sum of absolute values is preferable because it is computationally cheaper without loss of performance.

Preliminary Step-Kernels: The 2-D convolution kernels typically used for texture discrimination are generated from the following set of one-dimensional convolution kernels.

L5 = [1 4 6 4 1]
E5 = [-1 -2 0 2 1]
S5 = [-1 0 2 0 -1]
W5 = [-1 2 0 -2 1]
R5 = [1 -4 6 -4 1]

Figure 4.11: One-dimensional convolution kernels of length five

The mnemonics above stands for level, edge, spot, wave, and ripple. Note that all kernels except L5 are zero-sum. In his dissertation, Laws also presents convolution kernels of length three and seven kernels.

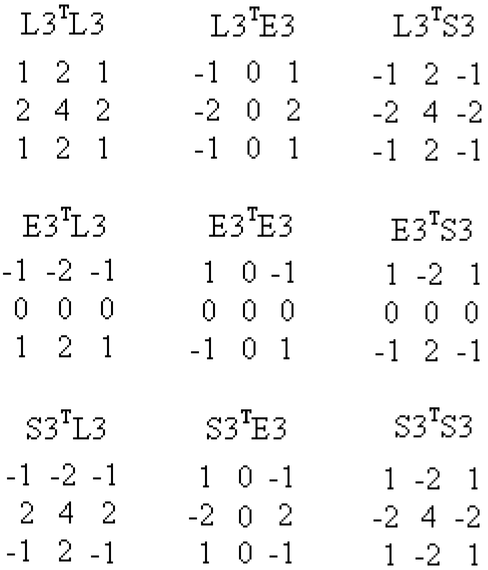
L3= [1 2 1]
E3 = [-1 O 1]
S3 = [-1 2 -1]

Figure 4.12: The1x3 kernels

L7=[1 6 15 20 15 6 1]
E7=[-1 -4 -5 – 5 4 1]
S7=[-1 -2 1 4 1 -2 -1]
W7=[-1 0 3 0 -3 0 1]
R7=[1 -2 -1 4 -1 -2 1]
O7=[-1 6 -15 20 -15 6 -1]

Figure 4.13: The 1x7 kernels

The two dimensional forms are generated from the one-dimensional forms by outer products. That is if k_1 and k_2 are two one-dimensional forms, each a row vector of K columns, then $k_1^t k_2$ constitutes a KxK kernel. For example, 3x3 kernels:



$L3^TL3$	$L3^TE3$	$L3^TS3$
1 2 1	-1 0 1	-1 2 -1
2 4 2	-2 0 2	-2 4 -2
1 2 1	-1 0 1	-1 2 -1

$E3^TL3$	$E3^TE3$	$E3^TS3$
-1 -2 -1	1 0 -1	1 -2 1
0 0 0	0 0 0	0 0 0
1 2 1	-1 0 1	-1 2 -1

$S3^TL3$	$S3^TE3$	$S3^TS3$
-1 -2 -1	1 0 -1	1 -2 1
2 4 2	-2 0 2	-2 4 -2
-1 2 -1	1 0 -1	1 -2 1

Figure 4.14: The nine 3 x 3 Laws' masks

Similarly by taking the inner product a vertical 1-D kernel of length 5 with a horizontal 1-D kernel of length 5, we can generate 25 different two-dimensional convolution kernels. Of the 25 two-dimensional convolution, 24 of them are zero-sum; the L5L5 kernel is not. Some Laws’ 5×5 center weighted masks are exampled below.

-1	-4	-6	-4	-1
-2	-8	-12	-8	-2
0	0	0	0	0
2	8	12	8	2
1	4	6	4	1

(a) horizontal edge

1	-4	6	-4	1
-4	16	-24	16	-4
6	-24	36	-24	6
-4	16	-24	16	-4
1	-4	6	-4	1

(b) high freq. spot

-1	0	2	0	-1
-2	0	4	0	-2
0	0	0	0	0
2	0	-4	0	2
1	0	-2	0	1

(c) *V*-shape

-1	0	2	0	-1
-4	0	8	0	-4
-6	0	12	0	-6
-4	0	8	0	-4
-1	0	2	0	-1

(d) vertical line

Figure 4.15: 5 x 5 Laws' masks examples

We now describe how to build up a set of texture energy measures for each pixel in a digital image.

Step I-Apply Convolution Kernels: Given a sample image with N rows and M columns that we want to perform texture analysis on, i.e. compute texture features at each pixel, we first apply each of our 25 convolution kernels to the image. The result is a set of 25 NxM greyscale images. These will form the basis for our textural analysis. Let $F_k[i,j]$ be the result of filtering with the kth mask at pixel [i,j]. Then the texture energy map E_k for filter k of 5x5 kernel is defined by

$$E_k[r,c] = \sum_{j=c-5}^{c+5} \sum_{i=r-5}^{r+5} | F_k[i,j] | \qquad (4.54)$$

Each texture energy map is a full image, representing the application of the kth mask to the input image.

Step II- Performing Windowing Operation: Having produced images that indicate local edginess, etc., we now want to replace every pixel in our 25 NxM separate greyscale images with a Texture Energy Measure (TEM) at the pixel is to deduce the local magnitudes of these quantities. We do this by looking in a local neighbourhood around each pixel and summing together the absolute values of the neighbourhood pixels. Laws used a 15 x 15 smoothing windows. We generate a new set of images, which we will refer to as the TEM images, during this stage of image processing. The following non-linear filter is applied to each of our 25 NxM images.

$$\mathrm{NEW}(x,y) = \underset{i=-7}{\overset{i=+7}{\mathrm{SUM}}} \ \underset{j=-7}{\overset{j=+7}{\mathrm{SUM}}} \ \mathrm{OLD}(x+i, y+j) \qquad (4.54)$$

Laws also suggests the use of another filter instead of the "absolute value windowing" filter listed above:

$$\mathrm{NEW}(x,y) = \mathrm{SQRT}(\underset{i=-7}{\overset{i=+7}{\mathrm{SUM}}} \ \underset{j=-7}{\overset{j=+7}{\mathrm{SUM}}} \ \mathrm{OLD}(x+i, y+j)^2) \qquad (4.55)$$

While Laws used both squared magnitudes and absolute magnitudes to estimate texture energy, the former corresponding to true energy and giving a better response, the latter are useful in requiring less computation.
We have at this point generated 25 TEM images from our original image. Lets denote these images by the names of the original convolution kernels with an appended ``T" to indicate that this is a texture energy measure, i.e. the non-linear filtering has been performed.

L5L5T	E5L5T	S5L5T	W5L5T	R5L5T
L5E5T	E5E5T	S5E5T	W5E5T	R5E5T
L5S5T	E5S5T	S5S5T	W5S5T	R5S5T
L5W5T	E5W5T	S5W5T	W5W5T	R5W5T
L5R5T	E5R5T	S5R5T	W5R5T	R5R5T

Figure 4.16: TEM images

Step III-Normalize Features for Contrast : All convolution kernels used thus far are zero-mean with the exception of the L5L5 kernel. In accordance with Laws' suggestions, we can therefore use this as a normalization image; normalizing any TEM image pixel-by-pixel with the L5L5T image will normalize that feature for contrast. After this is done, the L5L5T image is typically discarded and not used in subsequent textural analysis unless a ``contrast" feature is desirable.

Step IV- Combine Similar Features: For many applications, "directionality" of textures might not be important. If this is the case, then similar features can be combined to remove a bias from the features from dimensionality. For example, L5E5T is sensitive to vertical edges and E5L5T is sensitive to horizontal edges. If we add these TEM images together, we have a single feature sensitive to simple ``edge content".

Following this example, features that were generated with transposed convolution kernels are added together. We will denote these new features with an appended ``R" for ``rotational invariance".

E5L5TR = E5L5T + L5E5T

S5L5TR = S5L5T + L5S5T

W5L5TR = W5L5T + L5W5T

R5L5TR = R5L5T + L5R5T

S5E5TR = S5E5T + E5S5T

W5E5TR = W5E5T + E5W5T

R5E5TR = R5E5T + E5R5T

W5S5TR = W5S5T + S5W5T

R5S5TR = R5S5T + S5R5T

R5W5TR = R5W5T + W5R5T

Figure 4.17: Rotational invariant TEM images

To keep all features consistent with respect to size, we can scale the remaining features by 2:

E5E5TR = E5E5T*2

S5S5TR = S5S5T*2

W5W5TR = W5W5T*2

R5R5TR = R5R5T *2

Figure 4.18: Scaling TEM images

The result, if we assume we have deleted L5L5T altogether as suggested in Step III, is a set of 14 texture features which are rotationally invariant. If we stack these images up, we get a data set where every pixel is represented by 14 texture features.

Laws' method resulted in excellent classification accuracy quoted at, for example 87% compared with 72% for the cooccurrence matrix method, when applied to a composite texture image of grass, raffia, sand, wool, pigskin, leather, water and wood by Laws in1980. Research was undertaken by Pietikainen et al. in 1983 confirms that Laws' texture energy measures are more powerful than measures based on pairs of pixels i.e. cooccurrence matrices.

4.3.1.9. Local Binary Partition

Most texture approaches assume, either explicitly or implicitly, that the requested samples are identical to the database samples with respect to spatial scale, orientation, and gray scale properties. However, real-world textures can occur at arbitrary spatial resolutions and rotations and they maybe subjected to varying illumination conditions. In addition, the degree of computational complexity of most proposed texture measures is too high.

Very simple, but useful texture measure alternative is the local binary partition measure[34,1]. For the basic version of the LBP algorithm, for each pixel p in the image, the eight neighbours are examined to see if their intensity is greater than that of p. The results from the eight neighbours are used to construct an eight-digit binary number $b_1b_2b_3b_4b_5b_6b_7b_8$ where $b_i = 0$ if the intensity of the ith neighbour is less than or equal to that of p and 1 otherwise. A histogram of these numbers is used to represent the texture of the image. Two images or regions are compared by computing the L_1 distance between their n-bin histograms as defined :

$$L_1(H_1, H_2) = \sum_{i=1}^{n} | H_1[i] - H_2[i] | \tag{4.59}$$

Now, we'll study the LBP method in detail by defining a gray scale and rotation invariant operator. The proposed[34] texture operator below al1ows for detecting "uniform" local binary patterns at circular neighbourhoods of any quantisation of the angular space and at any spatial resolution.

The operator is derived for a general case based on a circularly symmetric neighbour set of P members on a circle of radius R, denoting the operator as $LBP_{P,R}^{riu2}$. Parameter P controls the quantisation of the angular space, whereas R determines the spatial resolution of the operator. The discrete occurrence histogram of the "uniform" patterns i.e., the responses of the $LBP_{P,R}^{riu2}$ operator, computed over an image or a region of the image is a very powerful texture feature. By computing the occurrence histogram, structural and statistical approaches are effectively combined: The local binary pattern detects microstructures e.g., edges, lines, spots, areas, whose underlying distribution is estimated by the histogram.

We start the derivation of our T gray scale and rotation invariant texture operator by defining texture T in a local neighbourhood of a monochrome texture image as the joint distribution of the gray levels of P(P > 1) image pixels:

$$T=t(g_c,g_0,...,g_{p-1}), (1) \tag{4.60}$$

where gray value g_c corresponds to the gray value of the centre pixel of the local neighbourhood and g_p(p =0, ..., P -1) correspond to the gray values of P equally spaced pixels on a circle of radius R (R > 0) that form a circularly symmetric neighbour set. If the coordinates of g_c are (O, O), then the coordinates of g_p are given by

$$(-R\sin(2\pi p/P), R\cos(2\pi p/P)) \tag{4.61}$$

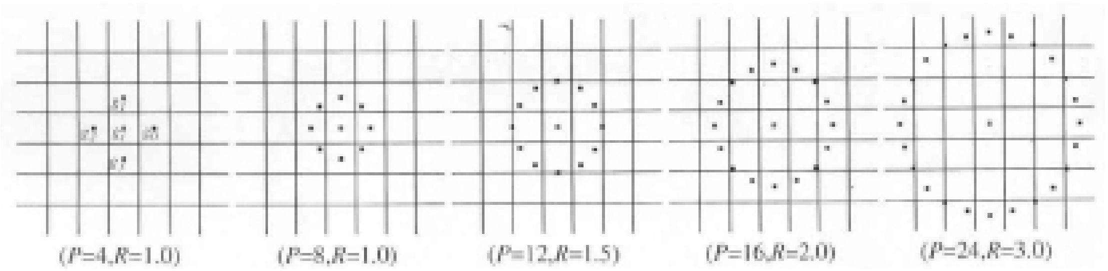


Figure 4.19: Circularly symmetric neighbour sets for different (P, R)

Fig. 4.19 illustrates circularly symmetric neighbour sets for various (P, R). The gray level values of neighbours, which do not fall exactly in the centre of pixel, are estimated by interpolation.

As the first step toward gray-scale invariance, we subtract, without losing information, the gray value of the centre pixel (g_c) from the gray values of the circularly symmetric neighbourhood g_p (p = 0,..., P -1), giving:

$$T = t(g_c, g_o-g_c, g_1-g_c, ..., g_{P-1}-g_c) \tag{4.62}$$

Next, we assume that differences g_p-g_c are independent of g_c, which allows us to factorise the Equation (4.59) :

$$T\approx t(g_c)t(g_0 -g_c,g_1-g_c,... ,g_{P-1} -g_c) \tag{4.63}$$

In practice, an exact independence is not warranted; hence, the factorised distribution is only an approximation of the joint distribution. However, the possible small loss in information is willingly accepted as it allows us to achieve invariance with respect to shifts in gray scale. Namely, the distribution $t(g_c)$ in the Equation (4.60) describes the overall luminance of the image, which is unrelated to local image texture and, consequently, does not provide useful information for texture analysis. Hence, much of the information in the original joint gray level distribution the Equation (4.58) about the textural characteristics is conveyed by the joint difference distribution $T \approx t(g_0 - g_c, g_1 - g_c, \ldots, g_{P-1} - g_c)$. This is a highly discriminative texture operator. It records the occurrences of various patterns in the neighbourhood of each pixel in a P-dimensional histogram. For constant regions, the differences are zero in all directions on a slowly sloped edge, the operator records the highest difference in the gradient direction and zero values along the edge and, for a spot, the differences are high in all directions.

Signed differences $g_p - g_c$ are not affected by changes in mean luminance; hence, the joint difference distribution is invariant against gray-scale shifts. We achieve invariance with respect to the scaling of the gray scale by considering just the signs of the differences instead of their exact values:

$$T \approx t(s(g_o - g_c), s(g_1 - g_c), \ldots., s(g_{P-1} - g_c)); (5) \tag{4.61}$$

where

$$s(x) = \{ 1, \ x \geq 0 \qquad 0, \ x < 0 \tag{4.62}$$

By assigning a binomial factor 2^p for each sign $s(g_p - g_c)$, we transform (5) into a unique $LBP_{P,R}$ number that characterizes the spatial structure of the local image texture:

$$LBP_{P,R} = \sum_{p=0}^{P-1} s(g_p - g_c) 2^p \tag{4.63}$$

The name "Local Binary Pattern" reflects the functionality of the operator, i.e., a local neighbourhood is thresholded at the gray value of the centre pixel into a binary pattern. $LBP_{P,R}$ operator is by definition invariant against any monotonic transformation of the gray scale, i.e., as long as the order of the gray level values in the image stays the same, the output of the $LBP_{P,R}$ operator remains constant.

The $LBP_{P,R}$ operator produces 2^P different output values, corresponding to the 2^P different binary patterns that can be formed by the P pixels in the neighbour set. When the image is rotated, the gray values g_p will correspondingly move along the perimeter of the circle around g_o. Since g_o is always assigned to be the gray value of element (0 , R) to the right of g_c rotating a particular binary pattern naturally results in a different LBP$_{P,R}$ value. This does not apply to patterns comprising of only 0s (or 1s) which remain constant at all rotation angles. To remove the effect of rotation, i.e., to assign a unique identifier to each rotation invariant local binary pattern we define:

$$LBP_{P,R}^{ri} = \min\{ROR(LBP_{P,R}, i) \quad | \quad i = 0,1,...,P-1\} \qquad (4.64)$$

where ROR(x, i) performs a circular bit-wise right shift on the P-bit number x i times.

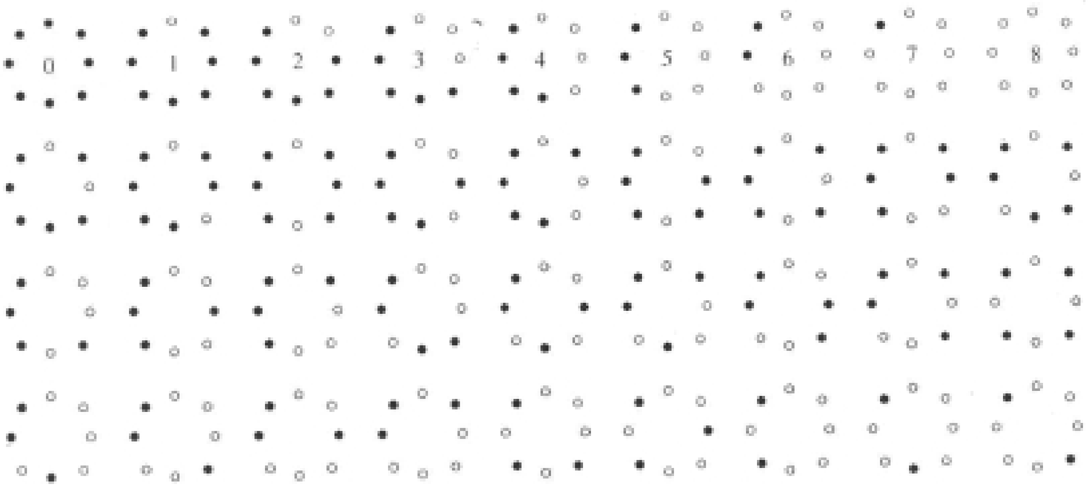


Figure 4.20: Rotation invariant binary patterns

There are 36 unique rotation invariant binary patterns that can occur in the circularly symmetric neighbour set of $LBP_{P,R}^{riu2}$. Black and white circles correspond to bit values of 0 and 1 in the 8-bit output of the operator. The first row contains the nine "uniform" patterns and the numbers inside them correspond to their unique $LBP_{P,R}^{riu2}$ codes.

In terms of image pixels, (8) simply corresponds to rotating the neighbour set clockwise so many times that a maximal number of the most significant bits, starting from g$_{P-1}$, is 0.

$LBP_{P,R}^{riu2}$ quantifies the occurrence statistics of individual rotation invariant patterns corresponding to certain micro features in the image; hence, the patterns can be considered as feature detectors. Figure 4.20 illustrates the 36 unique rotation invariant local binary patterns that can occur in the cage of P = 8, i.e., $LBP_{P,R}^{riu2}$ can have 36 different values. For example, pattern #0 detects bright spots, #8 dark spots and flat areas, and #4 edges. If we set R = 1, $LBP_{P,R}^{riu2}$ corresponds to the gray-scale and rotation invariant operator that we recall as LBP ROT .

Practical experience, however, has shown that LBP ROT as such does not provide very good discrimination. There are two reasons:

The occurrence frequencies of the 36 individual patterns incorporated in LBP ROT vary greatly and the crude quantisation of the angular space at 45° intervals.

Certain local binary patterns are observed to be fundamental properties of texture, providing the vast majority, sometimes over 90 percent, of all 3 x 3 patterns present in the observed textures. We call these fundamental patterns "uniform" as they have one thing in common, namely, uniform circular structure that contains very few spatial transitions. "Uniform" patterns are illustrated on the first row of Figure 4.20. They function as templates for microstructures such as bright spot (0), flat area or dark spot (8), and edges of varying positive and negative curvature (1-7).

To formally define the "uniform" patterns, a uniformity measure U ("pattern") is introduced, which corresponds to the number of spatial transitions, bitwise 0/1 changes, in the "pattern". For example, patterns 00000000_2 and 11111111_2 have U value of 0, while the other seven patterns in the first row of Figure 4.20 have U value of 2 as there are exactly two 0/1 transitions in the pattern. Similarly, the other 27 patterns have U value of at least 4. We designate patterns that have U value of at most 2 as "uniform" and propose the following operator for gray-scale and rotation invariant texture description instead of $LBP_{P,R}^{riu2}$:

$$LBP_{P,R}^{riu2} = \begin{cases} \sum_{p=0}^{P-1} s\left(g_p - g_c\right) & if \quad U\left(LBP_{P,R}\right) \leq 2 \\ P+1 & otherwise, \end{cases} \quad \text{where} \qquad (4.65)$$

$$U\left(LBP_{P,R}\right) = \left| s\left(g_{P-1} - g_c\right) - s\left(g_0 - g_c\right) \right| + \sum_{p=1}^{P-1} \left| s\left(g_P - g_c\right) - s\left(g_{p-1} - g_c\right) \right| \quad (4.66)$$

Superscript ^{riu2} reflects the use of rotation invariant "uniform" patterns that have U value of at most 2.

By definition, exactly P + 1 "uniform" binary patterns can occur in a circularly symmetric neighbour set of P pixels. Equation (4.65) assigns a unique label to each of them corresponding to the number of "1" bits in the pattern (O⇒P), while the "non-uniform" patterns are grouped under the "miscellaneous" label (P + 1). in Figure 4.20, the labels of the "uniform" patterns are denoted inside the patterns. In practice, the mapping from $LBP_{P,R}$ to $LBP_{P,R}^{riu2}$, which has P + 2 distinct output values, is best implemented with a lookup table of 2^P elements.

The final texture feature employed in texture analysis is the histogram of the operator outputs, pattern labels, accumulated over a texture sample. The reason why the histogram of "uniform" patterns provides better discrimination in comparison to the histogram of all individual patterns comes down to differences in their statistical properties. The relative proportion of "non-uniform" patterns of all patterns accumulated into a histogram is small that their probabilities cannot be estimated reliably. Inclusion of their noisy estimates in the dissimilarity analysis of sample and model histograms would deteriorate performance.

We noted earlier that the rotation invariance of LBPROT($LBP_{8,1}^{ri}$) is hampered by the crude 45° quantization of the angular space provided by the neighbour set of eight pixels. A straight forward fix is to use a larger P since the quantization of the angular space is defined by (360° / P). However, certain considerations have to be taken into account in the selection of P. First, P and R are related in the sense that the circular neighbourhood corresponding to a given R contains a limited number of pixels (e.g., nine for R = 1), which introduces an upper limit to the number of non-redundant sampling points in the neighbourhood. Second, an efficient implementation with a look-up table of 2^P elements sets a practical upper limit for P. P values up to 24, which requires a look-up table of 16 MB that can be easily managed by a modern computer.

The LBP method is manageable in real-world textures that can be subject to any arbitrary spatial resolution, rotation or varying illumination. Besides, its computational complexity is quite low. Therefore, LBP became one of the popular methods of the literature.

4.3.1.10. Frequency Based Spectral Approach

The psychophysical results indicate[4,5,20] that the human visual system analyses the textured images by decomposing the image into its frequency and orientation components; therefore, frequency based spectral approaches[37,38,8,16,9,36] are quite popular in the literature. Frequency-based methods rely on the analysis of the power spectral density function through the Fourier domain transform which may be computed in real time.

There are three features of the Fourier spectrum that are useful for texture description: (1) Prominent peaks in the spectrum give the principal direction of the texture patterns. A peak in the power spectrum represents a periodical pattern in the original spatial domain. Since the power spectrum rotates according to the spatial image's orientation, the orientation of the pattern is given by the direction of the harmonic frequency, closest to the origin. Images including non-periodical or random patterns have a power spectrum in which peaks are not easy to detect. (2) The location of the high-energy, narrow peaks in the frequency plane gives the fundamental spatial period of the patterns. Since the degree of texture coarseness is proportional to spatial period, coarse textures have spectral energy concentrated at low spatial frequencies. Fine textures, instead concentrate at high spatial frequencies. (3) Eliminating any periodic components via filtering leaves non-periodic image elements, which can then be described by statistical techniques. Average values of energy in specific wedges and rings of the Fourier spectrum, figure 4.21, can be used as textural description features. Features evaluated from rings reflect coarseness of the texture -high energy in large radius rings is characteristic of fine textures, high frequencies, while high energy in small radii is characteristic of coarse textures with lower spatial frequencies. Features evaluated from wedge slices of the Fourier transform image depend on directional properties of textures, if a texture has many edges or lines in a direction ϕ, high energy will be present in a wedge in direction ϕ + π/2.

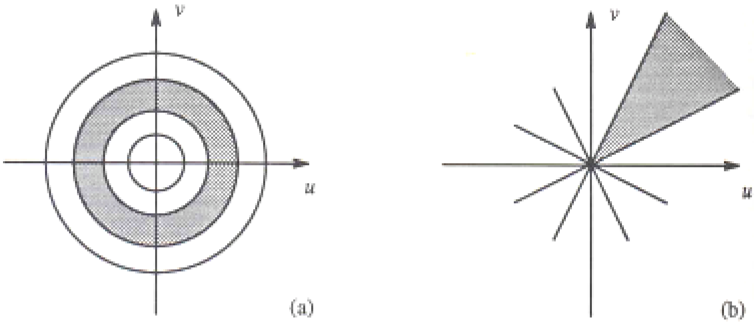


Figure 4.21: Partitioning of Fourier power spectrum: (a) ring filter (b) wedge filter

Detection and interpretation of the spectrum features just mentioned often are simplified mathematically by expressing the spectrum in polar coordinates to yield a function S(r,θ), where S is the spectrum function and r and θ are the variables in this coordinate system. For each direction θ, S(r,θ) may be considered a 1-D function

$S_\theta(r)$. Similarly, for each frequency r, $S_r(\theta)$ is a 1-D function. Analysing $S_\theta(r)$ for a fixed value of θ yields the behaviour of the spectrum, such as the presence of peaks, along a radial direction from the origin, whereas analysing $S_r(\theta)$ for a fixed value of r yields the behaviour along a circle centred on the origin.

A more global description is obtained by integrating these functions:

$$S(r) = \sum_{\theta=0}^{\pi} S_\theta(r) \tag{4.67}$$

$$S(\theta) = \sum_{\theta=0}^{R_0} S_r(\theta) \tag{4.68}$$

where R_0 is the radius of a circle centred at the origin.

The results of the equations above constitute a pair of values [S(r), S(θ)] for each pair of coordinates (r, θ). By varying these coordinates, we can generate two 1-D functions, S(r) and S(θ), that constitute a spectral-energy description of a texture for an entire image or region under consideration. Furthermore, descriptors of these functions themselves can be computed in order to characterise their behaviour quantitatively. Descriptors typically used for this purpose are the location of the highest value, the mean and variance of both the amplitude and axial variations, and the distance between the mean and the highest value of the function.

Similarly, a discrete image transform may be used for texture description. In this method, textured image is usually divided into small square non-overlapping sub-images. If the sub-image size is n x n , the grey levels of its pixels may be interpreted as a n^2-dimensional vector, and an image can be represented as a set of vectors. These vectors are transformed applying a Fourier transform. Maxima of the spectrum can be used as parameters for modelling texture properties. We implemented such a similar algorithm in which the image is partitioned into 32x32 sub-windows and the Fast Fourier Transform is taken for each piece. Then, each sub-window is further divided into 4x4 blocks energies that build the elements of the feature vector for that sub-window.

Spatial frequency texture description methods are based on a well-known approach. Despite that, many problems remain, the resulting description is not invariant even to monotonic image grey level transforms; further, Weszka et al. claims that the frequency-based approach is less efficient than others. Perhaps more important is the fact that the Fourier approach is a global one which is difficult to apply successfully to an image that is to be segmented by texture analysis and strong edges and image boundary effects can prevent accurate texture analysis A joint spatial/spatial-frequency approach is recommended.

4.3.1.11. Markov Random Field Models

The brightness level at a point in an image is highly dependent on the brightness levels of neighbouring points unless the image is simply random noise. In this section, we explain a noise model of this dependence, called the Markov random field[2,4,20,38,21,31].

The study of Markov random fields has had a long history, beginning with Ising's 1925 thesis on ferromagnetism. The model has traditionally been applied to the case of either Gaussian or binary variables, on a lattice. Besag allowed a natural extension to the case of variables that have integer ranges, either bounded or unbounded. This extension, coupled with estimation procedures, permit the application of the Markov random field to various image processing fields such as texture synthesis, texture classification, image segmentation, image restoration, and image compression. The Markov random-field model for textures assumes that the texture field is stochastic, stationary, and satisfies a conditional independence assumption. Let RxC be the spatial domain of an image, and for any (r,c) $\in$ RxC let N(r, c) denote the neighbours of (r, c).

The stationarity means that the spatial neighbourhood N(r,c) configuration is the same all over the image. This assumption is not valid at pixels near the image boundary. It is usual to assume the image is wrapped around a torus.

The conditional independence assumption is that the conditional probability of the pixel given all the remaining pixels in the image is equal to the conditional probability of the pixel given just the pixels in its neighbourhood,

$$\begin{aligned} &P\,(f\,(r, c) \mid f\,(i, j) : (i, j) \in R \times C, (i, j) \neq (r, c)) \\ &= P\,(f\,(r, c) \mid f\,(i, j) : (i, j) \in N\,(r, c)) \qquad (4.69) \end{aligned}$$

When the distributions of { P (f (r, c) : (r, c) $\in$ RxC) } are Gaussian, the discrete Gauss-Markov field can be written as an equation in which each pixel's value is a linear combination of the values in its neighbourhood plus a correlated noise term.

$$f(r,c) = \sum_{(i,j)\in N(0,0)} f(r-i,c-j)h(i,j) + u(r,c) \tag{4.70}$$

where the coefficients of the linear combination are given by the function h and { u(r, c) | (r, c) ∈ RXC } represent a joint set of possible correlated Gaussian random variables.

In the Markov model of the texture analysis, the intensity at any pixel depends only upon the intensity of the adjacent pixel in a neighbourhood and upon a transition probability matrix. The brightness level at a point in an image is highly dependent on the brightness levels of the neighbouring points unless the image is simply random noise. Markov random fields use a precise model of this dependence, they are able to capture the local (spatial) contextual information in an image. Let *Y(i,j)* denote the brightness level at a point (*i,j*) on the *N* x *N* lattice *L*. For the sake of simplicity let us take the labelling of the *Y(i,j)* to be *Y(i)*, *i*=1,2,..., *M* where $M = N^2$. A colouring of lattice *L*, or a colouring of *L* with *G* levels, denoted by *X* is a function from the points of *L* to the set {0,1,...,*G*-1}. The point *j* is said to be a neighbour of the point *i* if *P* (*Y(i)* | *Y(1), Y(2),..., Y(i-1), Y(i+2),..., Y(M)*) depends on *Y(j)*. Note that this definition does not imply that the neighbors of a point are necessarily close in terms of distance, although this is the usual case.

Now we can give the definition of a Markov random field:

A Markov random field is a joint probability density on the set of all possible colourings *Y* of the lattice *L* subject to the following conditions.

Positivity: *p*(*Y*) > 0 for all *Y*.

Markovianity: *P*(*Y(i)* | all point in the lattice except *i*) = *P*(*Y(i)*| neighbours of *i*).

Homogeneity: P(*Y(i)* | neighbours of *i*) depends only on the configuration of neighbours and is translation invariant (with respect to translates with the same neighbourhood configuration).

Let $N_p(i,j)$ denote the neighbourhood of the point (*i,j*). The neighbourhood $N_p(i,j)$ is shown in Figure 4.22 for *p* = 5. If we assume a unit distance between adjacent graph vertices then the first order MRF corresponds to a neighbourhood configuration of radius 1 that consists of the four nearest neighbours labelled by 1's, the second order MRF corresponds to a neighbourhood configuration of radius 2, that further includes the diagonal neighbours labelled by 2's, and so on. The neighbourhood $N_p(i,j)$ is shown in Figure 4.16 for *p* = 5.

5	4	3	4	5
4	2	1	2	4
3	1	X	1	3
4	2	1	2	4
5	4	3	4	5

Figure 4.22: Natural hierarchy of MRF models determined by neighborhood configurations of increasing order

As one could guess, the higher the order of the model is, the more powerful the model will be. For example a model of 0th order deals with only the pixel values, while a first-order model also encounters the relation between a pixel and its first-order neighbours.

Hence the GMRF is a non causal 2-D process described by the following equation

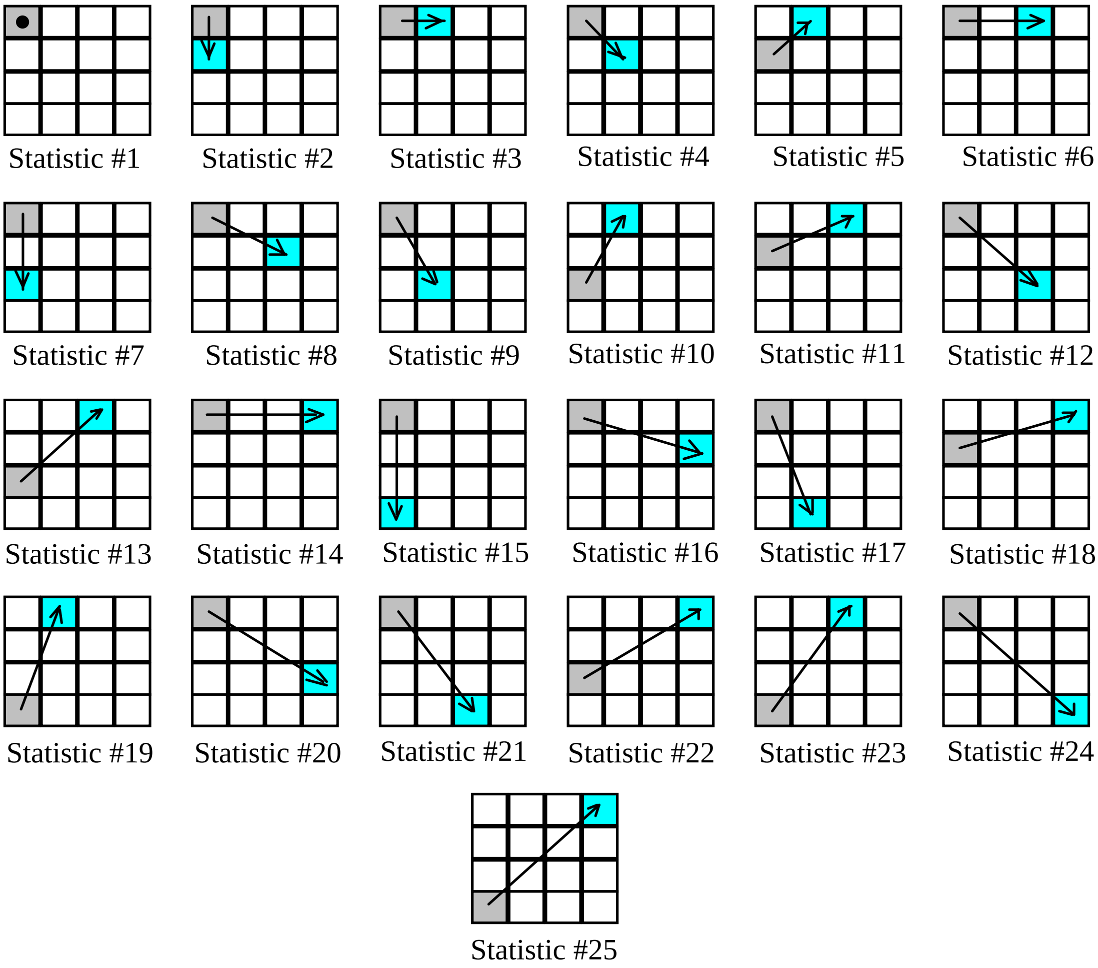
$$Y(r) = \sum_{v \in N_p(r)} \beta_{r-v} Y(v) + \varepsilon(r) \tag{4.71}$$

where $\varepsilon(r)$ is a Gaussian noise sequence with zero mean and the autocorrelation function given by

$$\begin{array}{ll} \sigma^2 & if\ v = r \\ -\sigma^2 / \beta_{r-v} & if\ v \in N_p \\ 0 & otherwise \end{array} \tag{4.72}$$

β_is are the parameters of the model to be estimated. However, estimating those parameters brings computational cost. Instead, sufficient statistics that define a parameter set may be estimated. Sufficient statistics are values that can describe a known model completely. For example for a Gaussian distribution σ and μ are the sufficient statistics. One can obtain the distribution if he only knows these two statistics.

Modelling a texture by using Markov random fields is not very difficult if sufficient statistics are used as model parameters instead of the actual model parameters. Cliques are used to determine the sufficient statistics. A clique is a graph whose vertex set is composed of vertices such that each one is a neighbour of all the others. The collection of cliques associated with a neighbourhood configuration defines a sufficient statistic for their corresponding MRF.



Statistic #1 Statistic #2 Statistic #3 Statistic #4 Statistic #5 Statistic #6

Statistic #7 Statistic #8 Statistic #9 Statistic #10 Statistic #11 Statistic #12

Statistic #13 Statistic #14 Statistic #15 Statistic #16 Statistic #17 Statistic #18

Statistic #19 Statistic #20 Statistic #21 Statistic #22 Statistic #23 Statistic #24

Statistic #25

Figure 4.23: Sufficient statistics of the ninth order model

Ninth-order models introduce 25 sufficient statistic values which are shown in Figure 4.23. For each pixel, the sufficient statistics are evaluated by multiplying the pixel gray level value, shaded dark gray in Figure 4.23, by the gray level value of the neighbouring pixel shown by the arrow, shaded light gray in Figure 4.23. After one calculates the sufficient statistics of a requested texture, he can use them to make a test to determine if the texture is similar enough to the requested texture. As one can see, all the statistics except for the first one, use pixels in the neighbourhood of a pixel thus requiring special care to be taken at the window boundaries. The window may either be considered to have a torroidal structure, or a mirror image of the window may be assumed to occur at the boundaries of the window, in order to deal with the boundaries properly. In the implementation, we preferred to omit the pixels near the boundaries that would force us to pass across the boundary while calculating a statistic. That means, all the pixels were used for the calculation of the first statistic whereas the pixels belonging to the right-most column were omitted for the calculation of the second statistic and so on. Although this seems as loss of data, using a torroidal structure may cause some disturbance due to the non-homogenous lighting effects.

MRF models are straightforward and easy to use and implement. Most of the textures may be modelled by them easily. MRF models are used not only for texture analysis but for texture synthesis also. Using sufficient statistics instead of the model parameters increase the time performance of the method. The calculated sufficient statistics may be regarded as feature values and used by any classification method based on feature vectors.

The biggest negative point is for natural images MRF is too weak a characterisation. Using Markov Random Field models requires the texture to have some properties described above. Textures violating some of those requirements may not be modelled using MRF models successfully. As the order of the model increases, the discrimination power increases, however, the number of sufficient statistics to be calculated also increases making the algorithm computationally more intensive. For finding the distance measure, calculation of the inverse of the covariance matrix may be required which also cause some trouble: For most textures, the matrix elements have very large values and the matrix is close to singularity. The precision of some computers may fail to calculate the inverse of the matrix correctly. Increasing the precision by using more precise data types, decrease the time performance.

4.3.2. Structural Approaches

Structural approaches[2,3,5,20,37,38,11] define the texture as "a set of primitive texels in some regular or repeated relationship". They assume that the texture is formed by texture primitives located via a placement rule. It can work well for man-made patterns but since most of the textures violate this assumption, the structural methods have limited power.

Suppose that we have a rule of the form S→aS, which indicates that the symbol S may be rewritten as aS. Three applications of this rule would yield the string aaaS. If a represents a circle as in the figure 4.24(a) and the meaning of "circles to the right" is assigned to a string of the form aaa ..., the rule S→aS allows generation of the texture pattern shown in Fig. 4.24(b).

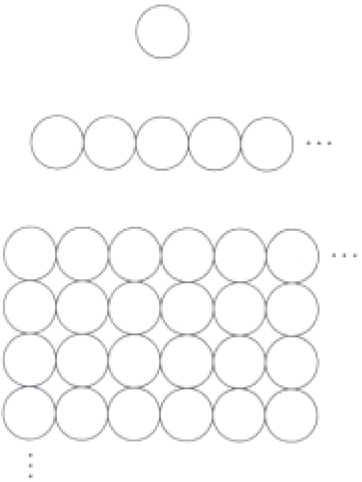


Figure 4.24: Texture patterns (a) texture primitive (b) pattern generated by the rule S→aS (c) 2D texture pattern generated by this and other rules.

Suppose next that we add some new rules to this scheme: S→bA, A→cA, A→c, A→bS, S→a, where the presence of a ‘b’ means "circle down" and the presence of a c means "circle to the left." We can now generate a string of the form aaabccbaa that corresponds to a 3x3 matrix of circles. Larger texture patterns, such as the one shown in Fig. 4.24(c), can be generated easily in the same way. The basic idea in the discussion made is that a simple "texture primitive" can be used to form more complex texture patterns by means of some rules that limit the number of possible arrangements of the primitive(s). The figure below shows that a simple primitive may generate various textures.

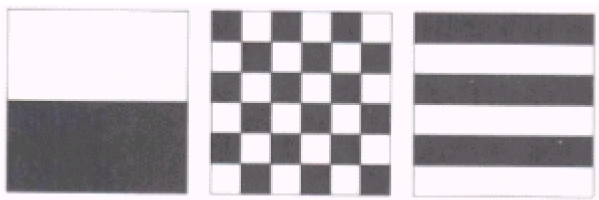


Figure 4.25: Three different textures with the same distribution of black and white: (a) Block pattern (b) Checkerboard (c) Striped pattern

Purely structural textures are deterministic texels, which repeat according to some placement rules, deterministic or random. In the case of deterministic placement rules, the spatial relationships may be expressed in terms of adjacency, closest distance, periodicities, and so on; the texture is labelled as being strong. For randomly placed texels, the associated texture is called weak and the placement rules may be expressed in terms of edge density, run lengths of maximal connected texels, relative extrema density, which is the number of pixels per unit area showing gray levels that are locally maxima or minima relative to their neighbours.

It is important to extract the texture elements (primitives) of a texture if we want to use structural methods. Usually texture elements consist of regions in the image with uniform gray levels. The texture primitives and the placement rules may be examined in spatial-frequency domain as well as the spatial domain. Given a texture primitive $h(x,y)$ and a placement rule $c(x,y)$, the texture $t(x,y)$ can be defined as

$$t(x,y) = h(x,y) * c(x,y) \tag{4.73}$$

where

$$c(x,y) = \sum \delta(x - x_m, y - y_n) \tag{4.74}$$

and x_m and y_n are the coordinates of impulse functions (the centres of the texture primitives located in the associated regions of the images). In the spatial-frequency domain

$$T(u,v) = H(u,v).C(u,v) \tag{4.75}$$

so that

$$C(u,v) = T(u,v).H^{-1}(u,v) \tag{4.76}$$

Thus, given a description of the texture primitive $h(x,y)$, we can derive a deconvolution filter $H^{-1}(u,v)$. Applying this filter to an image containing the texture of interest results in an array of impulses in the region of the image containing that texture. Each impulse is the centre of a texture primitive.

Purely syntactic (structural) methods of texture description are accompanied by many difficulties with syntactic analyser learning and with graph grammar inference. This is the main reason why purely syntactic methods are not widely used. On the other hand, a precise definition of primitives brings many advantages and it is not wise to avoid it completely.

4.3.3. Hybrid Approach

Hybrid methods[3] of texture description combine the statistical and syntactic approach. The technique is partly syntactic because the primitives are exactly defined, and partly statistical because spatial relations between primitives are based on probabilities.

Hybrid approach to texture description distinguishes between weak and strong textures. The syntactic part of a weak texture description focuses on extracting texture primitives on an image. Primitives can be described by their shape, size, etc. The simplest texture primitive is a pixel and its gray level property. Description of strong textures is based on the spatial relationships of texture primitives and two-directional interactions between primitives seem to carry most of the information. The texture description includes spatial relationships between primitives based on distance and adjacency relations. Using more complex texture primitives brings more textural information. On the other hand, all the properties of single pixel primitives are immediately available without the necessity of being involved in extensive primitive property computations.

In short, the hybrid system is based on primitive definition and spatial description of inter-primitive relations. Texture primitives are extracted first, and then based on recognised texture primitives, spatial relations between primitive classes are evaluated for each image.

4.4. Colour Algorithms

Besides the textural features studied, human vision utilises also colour information to judge similarity of objects. It is proven[4] by psychoanalysis that presence/absence of dominant colour, and degree of colourfulness are among strongest feature dimensions people use.

In the human eye, the overall perception is formed through the interaction of a luminance component, a chrominance component and an achromatic pattern component[21,13]. The luminance and chrominance components approximate signal representation in early visual cortical areas and extracts colour-based information while the achromatic pattern component approximates the signal formed at higher processing levels utilising textural pattern information. Similarly, for the machine vision, many combinational colour texture algorithms have been developed in the literature. Three fundamental colour based algorithms will be studied in the next section. But first, let us define the RGB colour space, which is our choice for colour representation throughout the thesis:

RGB colour space is formed to mimic the human colour processing. Eye has two different kinds of light receptors, called rods and cones. Rods do not respond to colour information. Cones are located in the central part of the eye, called the fovea. According to the colour range they respond, cones are divided into three subgroups called, cones responding to blue, cones responding to green, cones responding to red. Relative absorption ranges of these three cones are given in Figure 2.1.1.

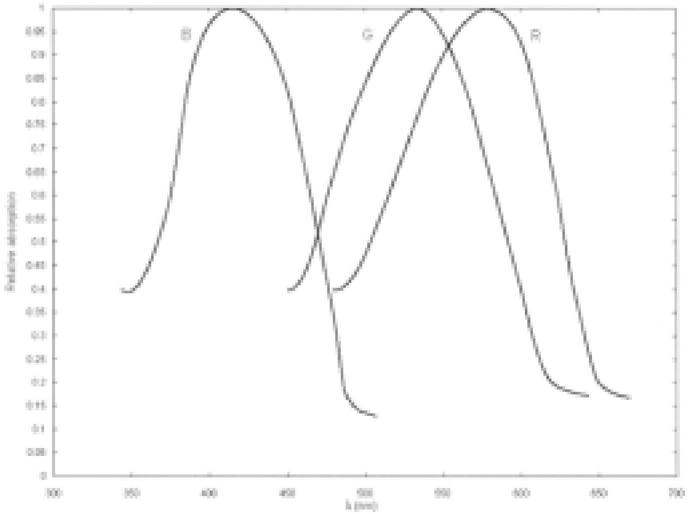


Figure 4.26: Relative absorption range of cones

RGB space is the most common colour representation but there are many other spaces developed for robust machine vision. A comparison of algorithms against different colour representations may be a future work of this study.

4.4.1. Colour Histograms

Evaluation of chromatic similarity, using colour histograms, can be performed by computing the L_1 or L_2 distances. The L_1 and L_2 distances between the query image histogram $H(I_Q)$ and the histograms of database images $H(I_D)$, are respectively defined[4,10] as:

$$D_H(I_Q, I_D) = \sum_{j=1}^{n} | H(I_Q, j) - H(I_D, j) | \tag{4.77}$$

$$D_H\left(I_Q, I_D\right) = \left(\sum_{j=1}^{n} \left(H\left(I_Q, j\right) - H\left(I_D, j\right) \right)^2 \right)^{1/2} \qquad (4.78)$$

where j is the index of the generic bin in the histogram. Most similar images are those that minimise these distances. Usually these metrics exhibit poor performance. Therefore the histogram intersection method is developed.

4.4.2. Histogram Intersection

Swain and Ballard have proposed⁴ histogram intersection for colour matching and indexing in large image database. Histogram intersection is defined as:

$$D_H\left(I_Q, I_D\right) = \frac{\sum_{j=1}^{n} \min\left(H\left(I_Q, j\right), H\left(I_D, j\right)\right)}{\sum_{j=1}^{n} H\left(I_D, j\right)} \qquad (4.79)$$

Colours that are not present in the query image do not contribute to the intersection distance. The number of bins in the histogram does not affect histogram intersection performance.

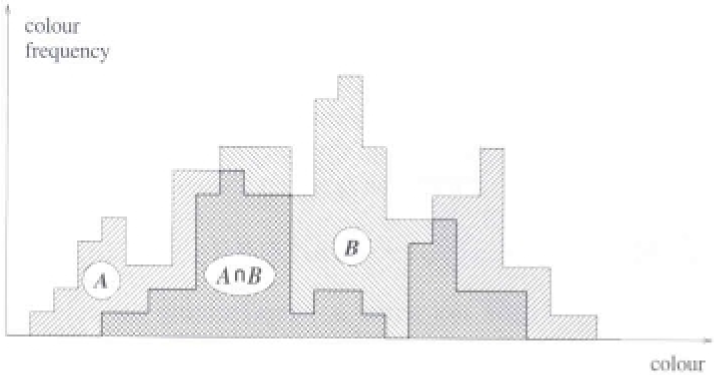


Figure 4.27: Histogram intersection between two histograms (labelled A and B)

Many applications require simple methods for comparing pairs of images based on their overall appearance. Colour histograms are a popular solution to this problem, and are used in the commercially available systems like QBIC and Chabot. Their advantages are efficiency, and insensitivity to small changes in camera viewpoint. However, colour histograms lack spatial information, it merely describes which colours are present in the image, and in what quantities and this can cause images with very different appearances to have similar histograms. For example, a picture of fall foliage might contain a large number of scattered red pixels; this could have a similar colour histogram to a picture of a single large red object. A histogram-based method for comparing images that incorporates spatial information is required. In addition, colour histograms are sensitive to both compression artefacts and camera auto-gain. Therefore, new algorithms have started to be seen in the literature trying to inherit the simplicity of the colour histograms but also adding new dimensions to them to compensate the disadvantages of the global metrics.

4.4.3. Colour Coherence Vectors

The colour's coherence is defined[13] as the degree to which pixels of that colour are members of large similarly coloured regions. These significant regions are referred as coherent regions, and they are of significant importance in characterising images. While a colour histogram counts the number of pixels with a given colour, a colour coherence vector (CCV) measures the spatial coherence of the pixels with a given colour. For example, the images below have similar colour histograms, despite their rather different appearances. The colour red appears in both images in approximately the same quantities. In the right image, the red pixels are widely scattered, while in the left image, the red pixels form a single coherent region.

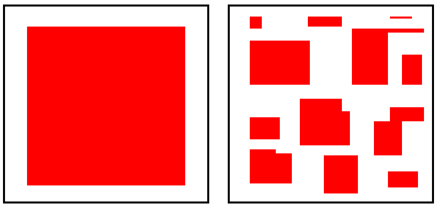


Figure 4.28: Coherent and non-coherent images

The initial stage in computing a CCV is similar to the computation of a colour histogram. First, the image is blurred slightly by replacing pixel values with the average value in a small local neighbourhood. Then, the colour space is discretized, such that there are only ‘n’ distinct colours in the image. The next step is to classify the pixels within a given colour bucket as either coherent or incoherent. A coherent pixel is part of a large group of pixels of the same colour, while an incoherent pixel is not. We determine the pixel groups by computing connected components. A connected component C is a maximal set of pixels such that for any two pixels $p,p' \in C$, there is a path in C between p and p'. Formally, a path in C is a sequence of pixels $p = p_1,p_2,., ,p_n = p'$ such that each pixel p_i is in C and any two sequential pixels p_i,p_{i+1} are adjacent to each other. We consider two pixels to be adjacent if one pixel is among the eight closest neighbours of the other; in other words, we include diagonal neighbours. When this step is complete, each pixel will belong to exactly one connected component. We classify pixels as either coherent or incoherent depending on the size in pixels of its connected component. A pixel is coherent if the size of its connected component exceeds a fixed value τ; otherwise, the pixel is incoherent. For a given discretized colour, some of the pixels with that colour will be coherent and some will be incoherent. Let us call the number of coherent pixels of the j^{th} discretized colour α_j and the number of incoherent pixels , β_j. Clearly, the total number of pixels with that colour is $\alpha_j+\beta_j$, and so a colour histogram would summarise an image as $<(\alpha_1+\beta_2),\ldots.,(\alpha_n+\beta_n)>$. Instead, for each colour we compute the pair (α_j,β_j) which we will call the coherence pair for the j^{th} colour. The colour coherence vector for the image consists of $<(\alpha_1 ,\beta_2),\ldots.,(\alpha_n ,\beta_n)>$. This is a vector of coherence pairs, one for each discretized colour.

Let us example the algorithm, to keep our example small, we will take τ= 4. Suppose that after we slightly blur the input image; the resulting intensities are as follows:

22	10	21	22	15	16
24	21	13	20	14	17
23	17	38	23	17	16
25	25	22	14	15	14
27	22	12	11	17	18
24	21	10	12	15	19

Figure 4.29: Average filtered input image

After discretizing the colour space so that bucket 1 contains intensities 10 through 19, bucket 2 contains 20 through 29, etc. , we obtain

2	1	2	2	1	1
2	2	1	2	1	1
2	1	3	2	1	1
2	2	2	1	1	1
2	2	1	1	1	1
2	2	1	1	1	1

Figure 4.30: Quantized input image

The next step is to compute the connected components. Individual components will be labelled with letters (A,B,.. .) and we will need to keep a table which maintaining the discretized colour associated with each label, along with the number of pixels with that label. Of course, the-same discretized colour can be associated with different labels if multiple contiguous regions of the same colour exist. The image may then become

B	C	B	B	A	A
B	B	C	B	A	A
B	C	D	B	A	A
B	B	B	A	A	A
B	B	A	A	A	A
B	B	A	A	A	A

Figure 4.31: Labelling the quantised image

and the connected components table will be

Label	A	B	C	D
Colour	1	2	1	3
Size	17	15	3	1

Figure 4.32: Connected Component Sizes

The components A and B have more than τ pixels, and the components C and D less than τ pixels. Therefore the pixels in A and B are classified as coherent, while the pixels in C and D are classified as incoherent.

A given colour bucket may thus contain only coherent pixels as does 2, only incoherent pixels as does 3, or a mixture of coherent and incoherent pixels as does 1.

The CCV for this image will be

Colour	1	2	3
α	17	15	0
β	3	0	1

Figure 4.33: CCV of the image

if we assume there are only 3 possible discretized colours, the CCV can also be written <(17,3),(15,0),(0,1)>.

Consider two images I and I', together with their CCV's G_I and $G_{I'}$, and let the number of coherent pixels in colour bucket j be α_j for I and α'_j for I'. Similarly, let the number of incoherent pixels be β_j and,β'_j. So

$$G_I = <(\alpha_1,\beta_2),\ldots.,(\alpha_n,\beta_n)> \qquad (4.80)$$

and

$$G_I = <(\alpha_1 ,\beta_2),\ldots.,(\alpha_n,\beta_n)> \qquad (4.81)$$

Colour histograms will compute the difference between I and I' as

$$\Delta H = \sum_{j=1}^{n} | (\alpha_j + \beta_j) - (\alpha_j^{'} + \beta_j^{'}) | \qquad (4.82)$$

The CCV method for comparing is based on the quantity

$$\Delta G = \sum_{j=1}^{n} | (\alpha_j - \alpha_j^{'} | + | \beta_j - \beta_j^{'} | \qquad (4.83)$$

From equations 1 and 2, it follows that CCV's create a finer distinction than colour histograms. A given colour bucket j can contain the same number of pixels in I as in I', i.e. $\alpha_j + \beta_j = \alpha'_j + \beta'_j$ but these pixels may be entirely coherent in I and entirely incoherent in I'. In this case $\beta_j = \alpha'_j = 0$, and while ΔH = 0, but ΔG will be large.

In general, ΔH ≤ ΔG. This can be proved by applying the triangle inequality:

$$\Delta H = \sum_{j=1}^{n} | (\alpha_j + \beta_j) - (\alpha_j^{'} + \beta_j^{'}) | \le \sum_{j=1}^{n} | (\alpha_j - \alpha_j^{'}) | + | (\beta_j - \beta_j^{'}) | = \Delta G \qquad (4.84)$$

Without normalisation, the distance between the coherence pairs (0,l) and (0,l00) is as large as the distance between (9000,900l) and (9900,9l00). Therefore, t is better to add a further normalisation step to ΔG.

The normalised difference between G_I and $G_{I'}$ is

$$\Delta = \sum_{j=1}^{n} \left| \frac{\alpha_j - \alpha_j^{'}}{\alpha_j + \alpha_j^{'} + 1} \right| + \left| \frac{\beta_j - \beta_j^{'}}{\beta_j + \beta_j^{'} + 1} \right| \qquad (4.85)$$

The denominators normalise these differences with respect to the total number of pixels. The factor of +l is used to avoid division by zero when the α's or β's are zero.

CHAPTER 5

IMPLEMENTATION AND RESULTS

This section summarizes the performance results obtained by implementing the texture algorithms discussed in the previous chapter. We'll first describe the database built, and define the features, distance metrics, and the performance criteria used. Next we tabulate and depict the results for the discussion.

5.1. Texture Databases

We have built a quite powerful image database including over 1000 static images in total. Each image is stored in jpeg format with the size of 128x128 pixels. Both gray scale and color samples are available, all of them RGB ordered.

The database composes of image groups which are derived from the famous texture benchmarking archives including MeasTex, VisTex, Outex, Broadatz, Rotated Brodatz and Columbia. The major problems to build a refined database are the scale dependency of the textures and the question of successful space coverage of the primitives:

We exclude natural scenes for our database and find the basic texture patterns, textels, that are unique and with enough differentiability power possibly to represent the whole texture world. This target may seem to be a bit unrealistic, but we see that as the sizes of the databases and benchmarking studies increase, more optimal image archives are reached. For this reason we tried to build one of the biggest image archives ever by using the proved databases available in the literature. To increase the complexity, we also included new images obtained from the Brown University marble archive, internet graphics pages, textile industry, and pictures taken at Sabancı University.

The performance of the algorithms are evaluated both on the subgroups and on the main database. Each subgroup has a bit different context and purpose. Brodatz is very widely used in the literature[4,10,27,28]. The rotated Brodatz reveals the performance response of the algorithms against transformations. Internet images emphasize the color nature of the problem. The textile section created shows whether the algorithms developed for this industry is content specific or not. Finally, the Brown database gives us an idea of how successfully we can use these algorithms on archaeological samples. We checked the algorithms against three databases: the Brodatz database, the archaelogical marbles and finally the big database built by including all the image groups mentioned above. You may find some representative samples of these three databases in Apendix B.

5.2. The System Designed

Our target is to create a CBIR Level 1 benchmarking platform for the texture and color algorithms available in the literature. The developed system focuses on the feature extraction step and includes the following algorithms which have been discussed in detail throughout the chapter 4: Autocorrelation Function, Markov Random Fields, Laws' Texture Energies, Color Coherence Vectors, Histogram Difference, Minimum Bin Histogram Difference, Fast Fourier Transform, Histogram Statistics, Linear Binary Patterns, Co-occurrence Matrices, Gray Level Variance Matrices, Gray Level Gap Length Matrices, Gray Level Run Length Matrices and Neighboring Gray Level Dependence Matrices.

The features used for each method are:

Histogram Statistics (HS): mean, standard deviation, normalized standard deviation, third moment, uniformity, entropy

Autocorrelation Function (AF): profile spreads, cross-relation, second-degree spread

Co-occurrence Matrices (CM): energy, entropy, contrast, homogeneity, correlation

Gray Level Variance Matrices (GLVM): smoothness, gray level variance ratio, minimum gray level, roughness, low variance emphasis, high variance emphasis, low gray level variance emphasis, high gray level variance emphasis

Markov Random Fields (MRF): clique matrices

Laws' Texture Energies (LTE): textural energy matrices

Neighboring Gray Level Dependence Matrices (NGLDM): small number emphasis, large number emphasis, number non-uniformity, entropy, second moment

Color Coherence Vectors (CCV): coherence vectors

Histogram Difference (HD): gray level histograms

Minimum Bin Histogram Difference (MBHD): gray level histograms

Fast Fourier Transform (FFT): quantized spectral energies

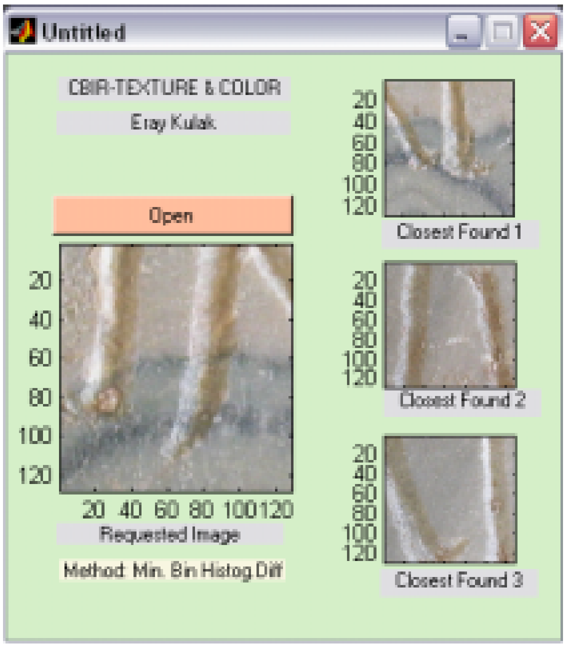
Linear Binary Patterns (LBP): 8 digit binary code histograms

Gray Level Gap Length Matrices (GLGLM): period metric, quasi-periodicity metrics

Gray Level Run Length Matrices (GLRLM): long primitives (run) emphases, short primitives emphasis, gray level uniformity, primitive length uniformity, and primitive percentage.

All the methods mentioned above require some preprocessing of the images before the main feature extraction codes run, RGB to greyscale conversion or color space channel filtering is accomplished first. Then the noise is removed and the macro features are emphasized by the gaussian, median or Wiener filters. For many codes, the number of grey levels is important for the performance. As the number gets higher, computations become more complex and probability of similarity between the compared pixels decreases. Quantization steps are added for these reasons. For the quantization, k-means clustering, uniform quantization and minimum variance quantization methods are implemented. Merging may be required if grouping of the similiar texels emphasize the general motif more. Standard “4-neighboured” , “8-neighboured” and “divide and conquer” algorithms are implemented for merging.

The system developed is a GUI based application. We used MATLAB interface design tools and I/O functions. In the GUI, the user easily selects the requested image and meanwhile the folder of the archive is marked. After selecting the required parameters, the sorting is accomplished by the code, and finally the closest 3 retrievals are displayed.



Method: Min. Bin Histog Diff

Figure 5.1: The GUI of the texture based CBIR system

For the similiarity distance metrics, Euclidean distance, City-block distance , Minkowsky distance , texture signatures and Mahalanobis distance are used.

Performance criteria of the retrieval are outlined in chapter 3 and results are in the following section.

The algorithms available in the literature are usually tested against 40-100 images for a general similarity analysis[4]. This is quite normal, because the CBIR systems available study just low level features. As seen in biological vision, higher level of semantic features are required for robust content retrievals. Therefore, for large databases it is better to find associations among features than solely depending on the primitive feature differentiability; otherwise exponential performance drops should be expected. A few current systems including QBIC, Virage work also on bigger environments including 1000 and over number of images. Even these systems don't work in real time, but store the feature data for a fast response against possible requests. Our archive therefore is a folder system which is an easy to build but hard to use system that works with real-time processing.

In the databases, each sample has four variants which build their class. If the retrieved image is from the same class of the requested, we count the case as correct retrieval. Such a decision criteria, adopted from machine vision, is not a ground–truth as explained in the previous chapters and furthermore may decrease the success rate because similiar images don't need to be from the same object according to the standart CBIR criteria. We undertake this burden because otherwise human subjective tests should be used which is not an objective test type. At the retrieval stage and performance criteria we never count the first retrieval which is surely the asked texel with zero distance.

One more final remark is, if a sample is associated with a textel in minimum distance, the reverse matching doesn't need to be expected always. So we requested from the platform each sample and recorded the whole order lists, therefore the usual report output of a single test for the biggest database is a 1000x1000 array.

5.3. Performance Results

5.3.1. Archaeological Marbles Database

This section explores the performances of the primitive feature extraction algorithms on databases with archaeological context. The samples are from the Brown University's marble archive. The need to follow the patterns on the neighbourhood zone of the broken marble pieces is a semantic level of problem as discussed. Nevertheless, in the figures and tables below, we try to get an idea how promising the metrics are, to be used for higher levels of algorithms. It should be taken account that we didn't build special environments controlling resolution, angle of view and lighting. Less freedom of such variables on texture sampling will surely increase the performances of the algorithms.

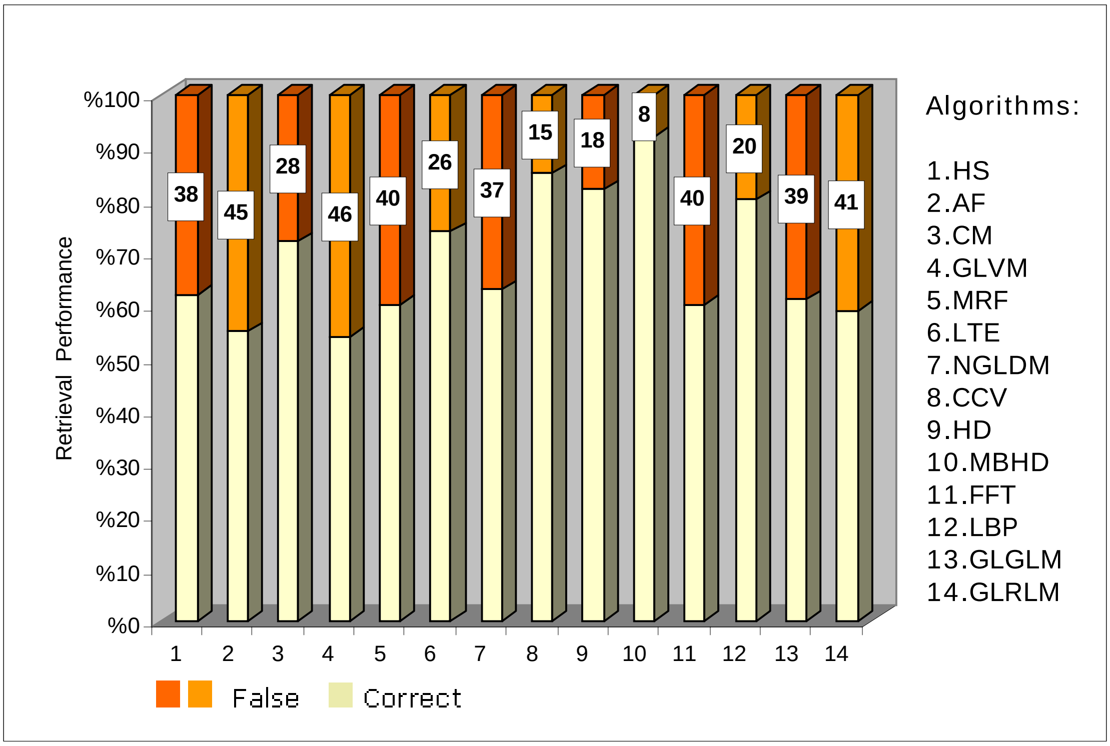


Figure 5.2: Cumulative recall-precision performance

As explained in section 5.2, each database sample has four variants, which build their class. If the retrieved image is from the requested class, we count the case as correct, otherwise as false retrieval. The overall performance is called 'recall-precision' as explained detailed in section 3.4.3. In Figure 5.2, it is clear that MBHD measure performs far better. This partially because our archaeological database doesn't include rotational variants of the samples. So, the value of the geometric invariant algorithms like CCV, LBP couldn't be understood easily on this archive.

Another observation is the colour feature metrics, used in HD, MBHD, and CCV, are quite powerful in general, probably because as explained in Chapter 3, the colour is the primary feature that the human vision utilises.

Finally, it should be cared that each algorithm focuses on different image features like frequency, colour, pixel gap length, etc. So they are not competitive but complementary algorithms. Combinational and higher level of algorithms may reduce the false acceptance by creating a higher dimensional feature space.

Retrieval performance in percentages

1-10	11-20	21-30	31-50	51-75	76-100	101-200	201-1000	Algorithms:
100	0	0	0	0	0	0	0	Ideal Case
64,5	20,5	14,7	0,1	0,2	0	0	0	1.HS
58,3	24,4	11	5,3	0,5	0,2	0,2	0,1	2.AF
75,1	10,8	6,9	4,4	1,7	0,4	0,5	0,2	3.CM
57,2	26,4	9,7	5,5	0,9	0	0,3	0	4.GLVM
62,3	20,5	8,5	3,3	2,5	1,7	0,9	0,3	5.MRF
76,8	19,5	2,5	1,1	0,1	0	0	0	6.LTE
67,4	22,3	0,5	4,2	2,6	1,5	0,7	0,8	7.NGLDM
89,1	10,2	0,3	0,2	0,2	0	0	0	8.CCV
86,3	10,1	2,3	1	0,1	0,2	0,1	0	9.HD
99,0	0,4	0,2	0,3	0,1	0	0	0	10.MBHD
62,8	22,4	8,5	3,2	1,2	0,8	0,6	0,5	11.FFT
82,7	10,3	4,2	2,1	0,4	0,3	0	0	12.LBP
63,3	20,5	11,8	2,6	1,7	0	0,1	0	13.GLGLM
62,8	28,4	6,3	1,4	0,9	0,1	0	0,1	14.GLRLM

Table 5.1: Cumulative distribution of the shortest interval lengths to retrieve all relevant

We tabulated above performances of the algorithms against the number of retrieved samples to see the distribution of the shortest interval lengths to retrieve all relevant images, the ones from the same class, over the database. The first row shows us the ideal distribution in which all correct retrievals have been counted in the closest ten distances.

Sharp decreases are seen in MBHD, CCV, HD, LBP methods which agree with the 'cumulative recall-precision performance' in Figure 5.2.

One more interesting observation is that the table shows all the algorithms are quite successful in decreasing the search space from thousands to tens of samples.

Retrieval performance in percentages

1-10	11-20	21-30	31-50	51-75	76-100	101-200	201-1000	Algorithms:
100	0	0	0	0	0	0	0	Ideal Case
91,5	6,6	1,7	0,1	0,1	0	0	0	1.HS
86	9,9	3,4	0,5	0,1	0	0	0	2.AF
97,5	1,4	0,9	0,2	0	0	0	0	3.CM
90,1	7,6	2,3	0	0	0	0	0	4.GLVM
96,3	3,1	0,5	0,1	0	0	0	0	5.MRF
96,8	2,2	0,4	0,2	0,1	0,1	0	0	6.LTE
93,2	4,7	1,5	0,3	0,3	0	0	0	7.NGLDM
99,5	0,2	0,1	0,1	0,1	0	0	0	8.CCV
97	1,5	0,9	0,4	0,2	0	0	0	9.HD
99,3	0,5	0,1	0,1	0	0	0	0	10.MBHD
94,6	5,1	0,1	0,1	0,1	0	0	0	11.FFT
97,3	1,2	0,8	0,4	0,2	0,1	0	0	12.LBP
95,4	2,3	1,9	0,3	0,1	0	0	0	13.GLGLM
89,7	8,4	1,2	0,4	0,2	0,1	0	0	14.GLRLM

Table 5.2: Cumulative distribution of the shortest interval lengths to retrieve first relevant

The measure, ‘cumulative distribution of the shortest interval lengths to retrieve first relevant’ explores how many images in average we should evaluate to include at least one correct class member for a request. If one accepts that getting a representative sample of the request’s class among first ten closest distances is enough, Table 5.2 in comparison with Table 5.1 shows that the most algorithms perform with over %95, i.e. reducing the search space significantly.

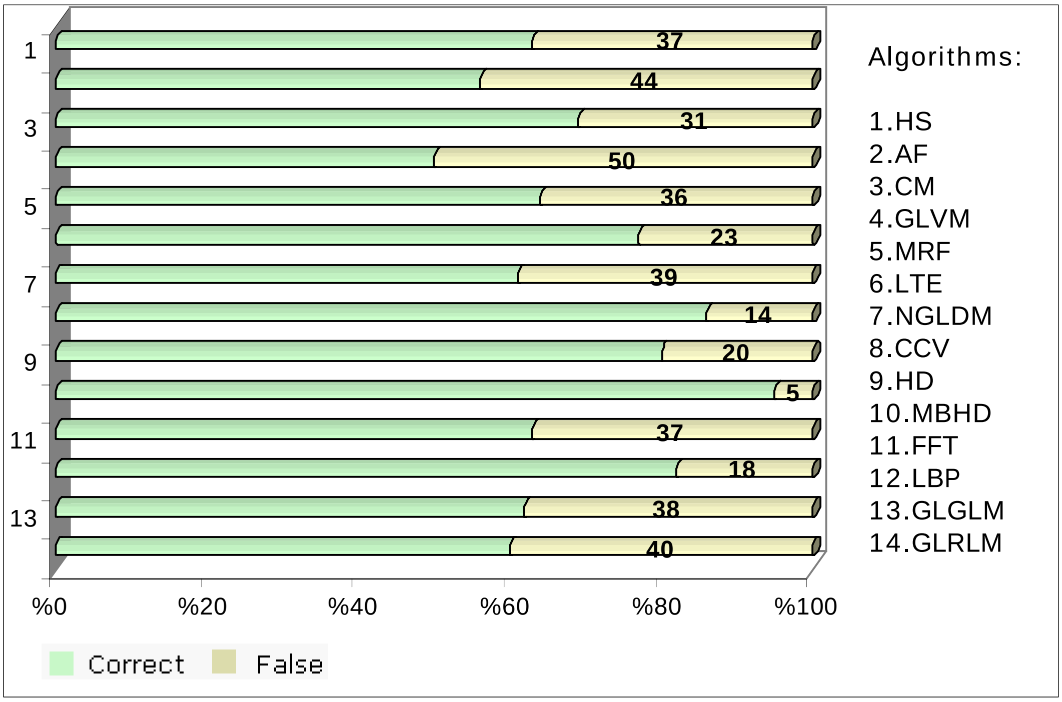


Figure 5.3: The cumulative correctness for the closest retrieval

The cumulative correctness for the closest retrieval demonstrates how successful we get the right class at the first choice. AF, GLVM, GLRLM are dependent on small rotational or translational transformations; therefore it is normal to observe such low performances with them.

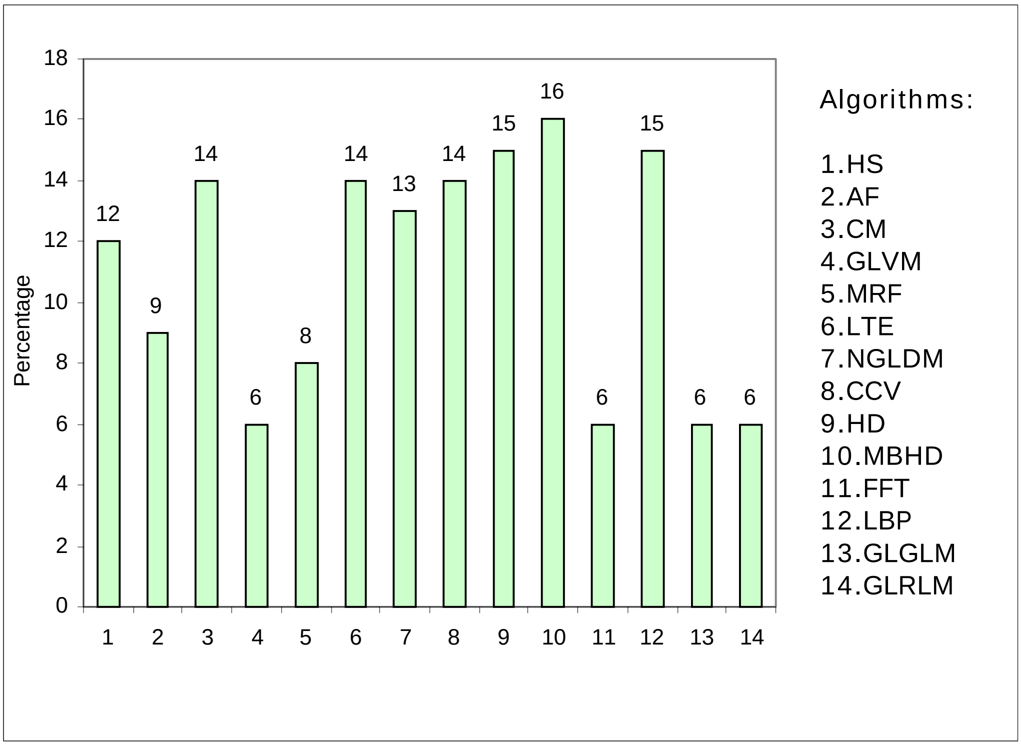


Figure 5.4: The distribution of the first three all correct retrievals

Because the database composes of four variants for each sample in the database, for each request, filling the first three closest places in distance with the relevant retrievals is the optimal result. The ‘percentage of the first three all correct retrieval’ measure explores the frequency of these optimal cases. Taking the previous figures and tables into account, the algorithms can be clustered into two groups with their success: LTE, CCV, HD, MBHD, LBP are better in performance because they focus more on global image features. Our textures are primitive space representatives; so, the success of CM, NGLDM, GLGLM and GLRLM methods, which count more the distributions of primitives, is less than the first group of algorithms.

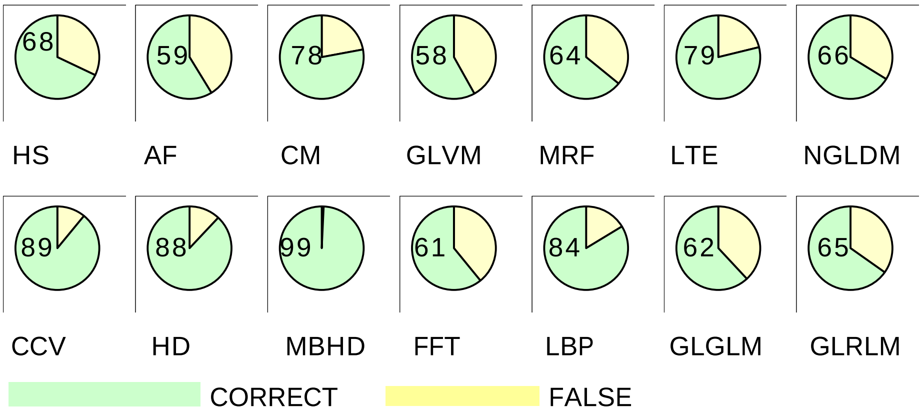


Figure 5.5: The percentage of the first three correct retrievals- majority rule applied

The majority rule is to take decision not only looking to the closest distance but taking a number of top distances into account. In the Figure 5.5., we are looking for the closest three similarities and decide whether the overall retrieval decision with majority rule applied represents the correct class for the request. In comparision with Figure 5.2, the correct retrieval pecentage of MBHD seems to increase from %92 to %99 by the majority rule. Although not always as signaficant as MBHD method, the other metrics perform better with the majority rule on decision. Also the performance clusters among the retrieval methods have been more seperated with this measure.

5.3.2. Brodatz

The Brodatz database is used almost as standard for benchmarking purposes in the literature. It is composed of 112 gray level primitive samples, which don't have a special context. The figures and tables below will give an idea for further studies to compare the algorithms developed with the non-textural features of the content based retrieval and similar textural feature extraction experiments done on Brodatz database.

In Figure 5.6, with comparison to Figure 5.2, the performances dropped a little. This is because the Brodatz database covers much more homogenous features although the textural feature space dimension is reduced by the missing colour information; the Brodatz samples are in gray scale and contain rotational variants of the samples. Therefore, the geometric invariant methods LTE, CCV, HD, MBHD and LBP are more successful.

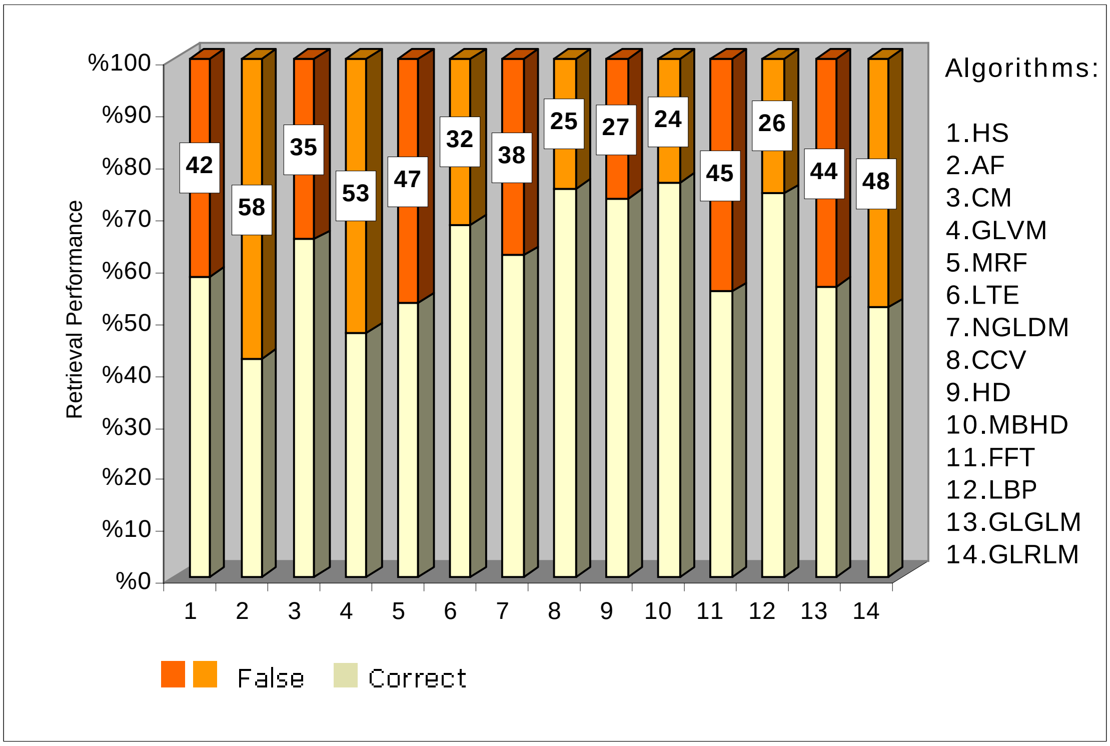


Figure 5.6: Cumulative recall-precision performance

Retrieved performance in percentages

1-10	11-20	21-30	31-50	51-75	76-100	101-200	201-1000	Algorithms:
100	0	0	0	0	0	0	0	Ideal Case
67	24,1	6,3	2,5	0	0,1	0	0	1.HS
55,4	26,2	10,4	5,5	2,1	0,2	0,2	0	2.AF
74,6	14,5	10,1	0,5	0,1	0,1	0	0,1	3.CM
56,3	22,9	8,4	7,6	4,3	0,3	0	0,2	4.GLVM
67,8	19,6	10,2	1,2	0,9	0	0,1	0,2	5.MRF
79,8	10,5	4,5	4	0,8	0,2	0,2	0	6.LTE
71,3	12,7	10,5	4,6	0,9	0	0	0	7.NGLDM
79,6	9,9	9,4	0,7	0,1	0,2	0,1	0	8.CCV
77,5	10,5	5,7	3,6	2,4	0	0,2	0,1	9.HD
83,9	11,1	3,1	1,3	0,2	0,1	0,1	0,2	10.MBHD
64,7	13,6	12,1	8,6	0,5	0,4	0	0	11.FFT
80,8	10,2	8,8	0,2	0	0	0	0	12.LBP
64	19,6	8,5	5,7	1,4	0,5	0	0,3	13.GLGLM
56,1	20,7	10,3	6,9	4,8	1	0,1	0,1	14.GLRLM

Table 5.3: Cumulative distribution of the shortest interval lengths to retrieve all relevant

As observed in Table 5.3, the MBHD, LBP, LTE, HD and CCV metrics reduce practically the search space from over 1000 images to approximately 30 retrieves. But even the HS, AF and FFT algorithms perform with over %99 success, in the top 75 retrieves.

Retrieved performance in percentages

1-10	11-20	21-30	31-50	51-75	76-100	101-200	201-1000	Algorithms:
100	0	0	0	0	0	0	0	Ideal Case
90,4	5,8	3,5	0,2	0,1	0	0	0	1.HS
85,1	12,6	2,1	0,2	0	0	0	0	2.AF
96,9	2,5	0,5	0,1	0	0	0	0	3.CM
89	10,4	0,4	0,2	0	0	0	0	4.GLVM
95,6	4,3	0	0,1	0	0	0	0	5.MRF
95,4	3,9	0,3	0,3	0,1	0	0	0	6.LTE
92,5	6,3	0,7	0,4	0,1	0	0	0	7.NGLDM
98,7	1,1	0,2	0	0	0	0	0	8.CCV
96,4	2,7	0,6	0,2	0,1	0	0	0	9.HD
98,8	0,5	0,5	0,2	0	0	0	0	10.MBHD
93,9	5,1	0,7	0,2	0,1	0	0	0	11.FFT
96,1	3,2	0,4	0,2	0,1	0	0	0	12.LBP
94,2	3,3	1,9	0,5	0,1	0	0	0	13.GLGLM
88,8	10,2	0,7	0,3	0	0	0	0	14.GLRLM

Table 5.4: Cumulative distribution of the shortest interval lengths to retrieve first relevant

Both of the Tables 5.2 and 5.4 show us the first ten retrievals are closely related with the requested image. Although the first interval performances slightly increase in Brodatz database, the distributions shifted more to the centre, ideal case, in archaeological database. This is because the Brodatz database's samples are homogenously distributed in feature space and they emphasize the geometric invariance of the algorithms which triggers confusions of rank among the closest retrievals.

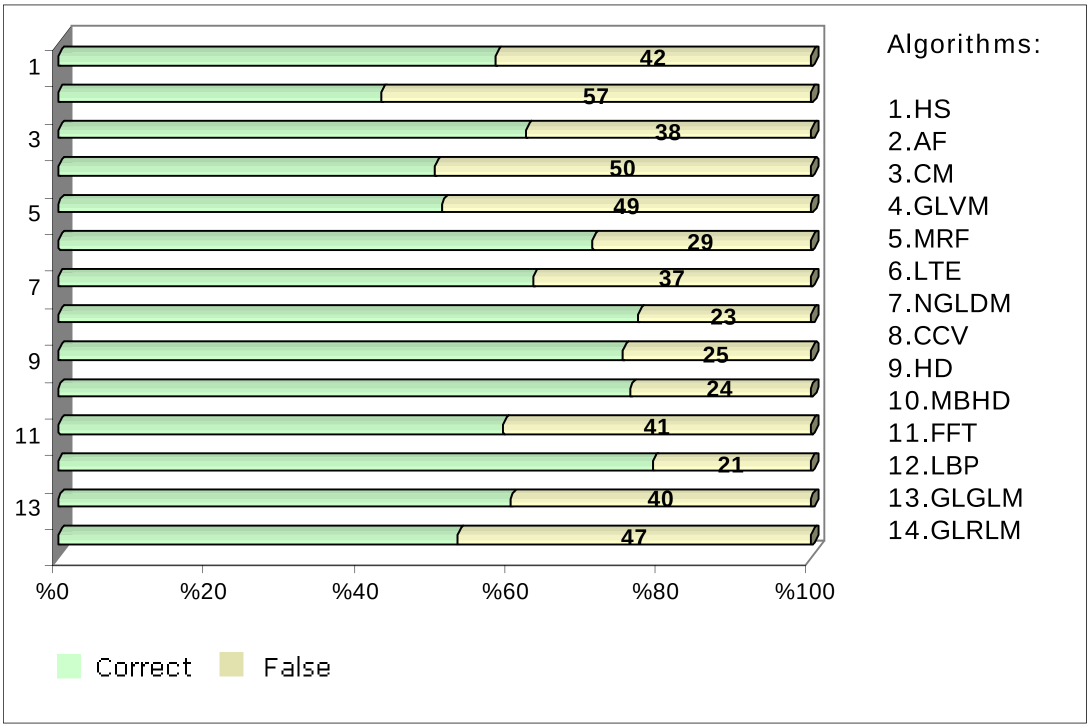


Figure 5.7: The cumulative correctness for the closest retrieval

In Figure 5.7, we observe that because the features are distributed homogenously in the Brodatz database, the performance results of LTE, CCV, HD, MBHD and LBP are almost the same. This shows once more that the algorithms work on independent feature axis in space. If the lighting, resolution and geometric transformation were controlled more strictly, some other algorithms like CM, GLVM, NGLDM, GLGLM and GLRLM would give similar results.

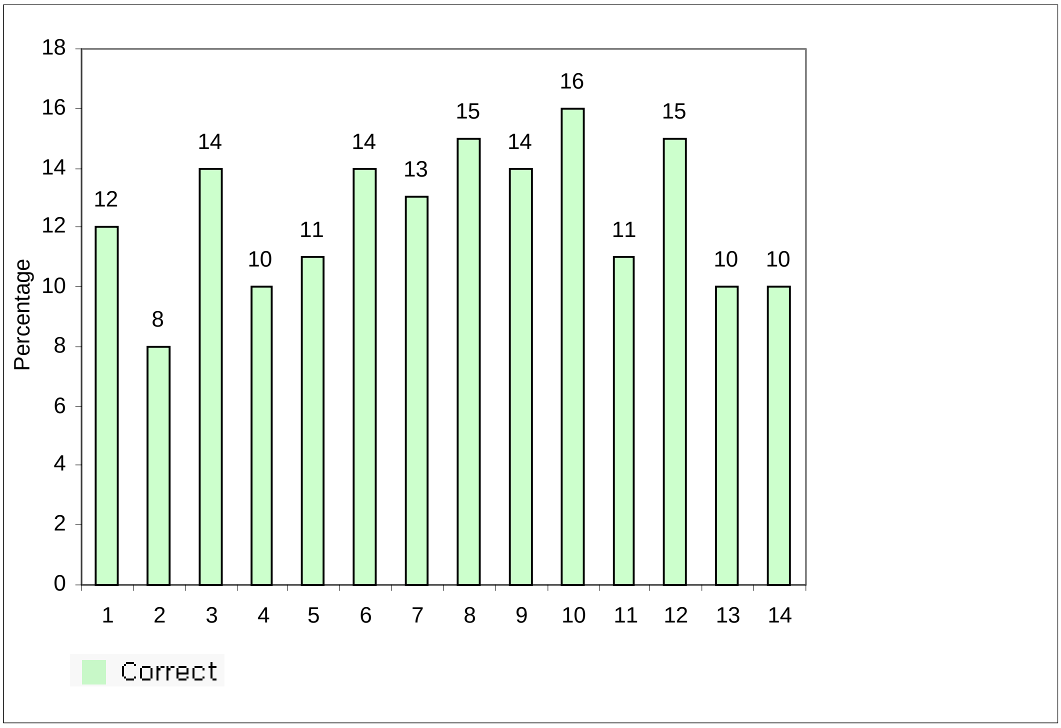


Figure 5.8: The distribution of the first three all correct retrievals

As observed in Figure 5.8, the performances are much closer to each other in Brodatz database. Each algorithm focuses mainly on a specific feature dimension. And because the textural variability is almost homogenous among Brodatz samples, it is hard to evaluate which feature dimension is more dominant.

A better approach would be to use combinatory metrics to work in a higher dimensional feature space.

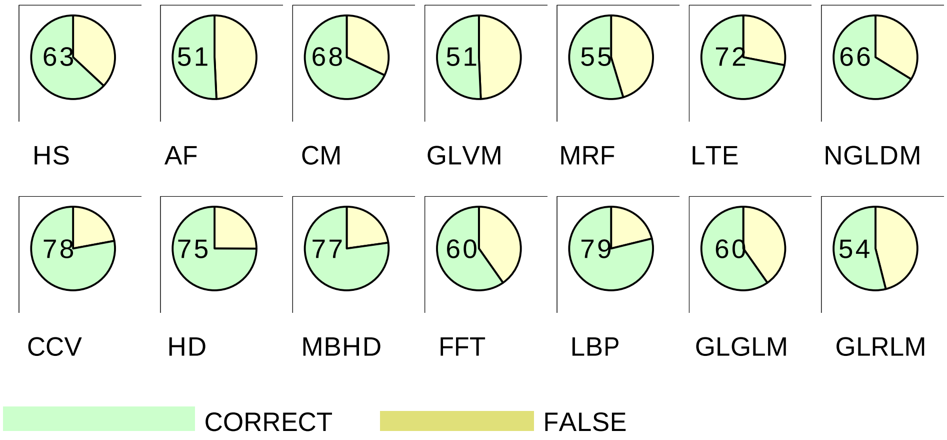


Figure 5.9: The percentage of the first three correct retrievals –majority rule applied

Although the order changes, the successful algorithms LTE, CCV, HD, MBHD and LBP preserve their superior performances also in Brodatz database. HS, AF, and FFT have known theoretical problems of differentiating similar textures; they may produce the same metrics for different samples. GLGLM, GLRLM, GLVM, NGLDM and CM could perform better if there weren't geometric translations on the database and the samples were not just primitive representatives.

5.3.3. The big database

The most challenging need for content-based retrieval is to preserve the performance, as the size of the database samples increases. Currently, because the primitive features are not discriminative enough, we don't see in the literature image archives with over 1000 samples. In this section, what we call as the 'big database' will demonstrate the rate of change in algorithms' discrimination power against database size with 1052 samples.

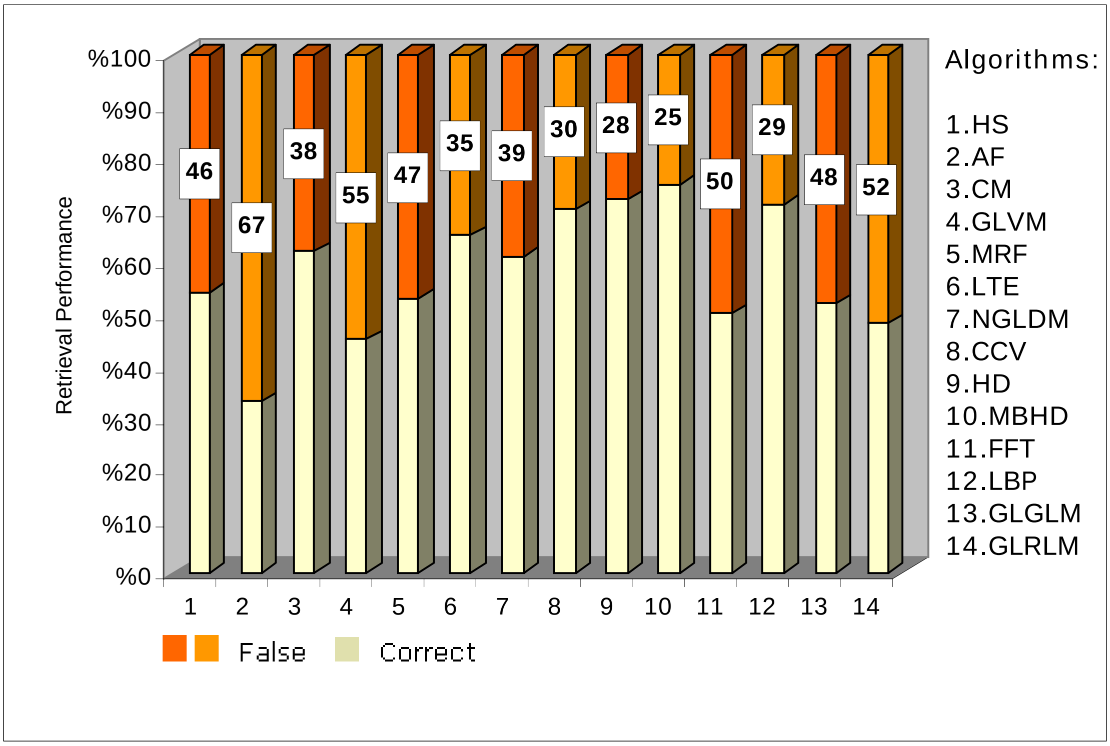


Figure 5.10: Cumulative recall-precision performance

First of all, in Figure 5.10, we see that the performance drop for AF is quite significant. This is because from its theory, different samples may have the same AF distances. The best performance is observed again by MDBH with %75, which is actually a highly promising result for using in semantic solutions. We would expect that LBP or CCV methods perform better than the MBHD because of known insufficiencies of histograms. However because it is highly difficult to create a homogenous database, probably we overemphasised the colour information but missed the pixel distribution differences as sampling the textures.

Retrieved performance in percentages

1-10	11-20	21-30	31-50	51-75	76-100	101-200	201-1000	Algorithms:
100	0	0	0	0	0	0	0	Ideal Case
61,2	24,9	9,5	3,1	1,2	0	0,1	0	1.HS
42,2	30,6	22,3	3,5	0,9	0,4	0	0,1	2.AF
70,3	16,4	12,4	0,7	0,2	0	0	0	3.CM
51,6	36,3	10,9	0,4	0,5	0,2	0,1	0	4.GLVM
64,3	20,4	9,4	3,1	1,1	0,9	0,5	0,3	5.MRF
72,6	14,6	6,5	3,4	1,7	0,5	0,4	0,3	6.LTE
69,4	20,3	7,1	1,2	1,3	0,6	0,1	0	7.NGLDM
76,3	16,2	5,4	0,9	0,7	0,3	0	0,2	8.CCV
74,5	18,6	4,2	2,6	0,1	0	0	0	9.HD
80,1	10,7	6,5	2,3	0,1	0,3	0	0	10.MBHD
60,3	20,1	15,3	3,5	0,5	0,2	0	0,1	11.FFT
75,2	20,9	1,4	1,1	0,8	0,4	0,2	0	12.LBP
59,9	34	5,5	0,2	0,1	0,1	0,1	0,1	13.GLGLM
50,4	40,2	3,9	2,5	1,2	0,9	0,5	0,4	14.GLRLM

Table 5.5: Cumulative distribution of the shortest interval lengths to retrieve all relevant

In comparison with the Tables 5.1 and 5.3, we observe the performances exponentially drop in the big database. Nevertheless, if we think that the advantage to use the automatic retrieval procedures is to downsize the search space, we see that the algorithms retrieve the first 50 similiar images including all the related samples with more than %98 success rate. Some algorithms like GLVM and GLRLM don’t perform well at filling the first ten ranks, but when looking the first 30 samples retrieved, they include the all relevant with over %90 success. Therefore, for content based retrieval, guessing the closest image as requested, like in a classification problem, is not always possible and also not much needed.

Retrieved performance in percentages

1-10	11-20	21-30	31-50	51-75	76-100	101-200	201-1000	Algorithms:
100	0	0	0	0	0	0	0	Ideal Case
80,9	19	0	0,1	0	0	0	0	1.HS
69,6	24,5	5,3	0,4	0,2	0	0	0	2.AF
89,9	9,5	0,3	0,2	0,1	0	0	0	3.CM
72	24,5	3,4	0,1	0	0	0	0	4.GLVM
83,7	14,3	1,7	0,3	0	0	0	0	5.MRF
93	3,8	2,2	0,6	0,4	0	0	0	6.LTE
89,6	10,1	0,2	0,1	0	0	0	0	7.NGLDM
95,8	3,9	0,2	0	0,1	0	0	0	8.CCV
93,9	4,8	1	0,3	0	0	0	0	9.HD
97	2,5	0,4	0,1	0	0	0	0	10.MBHD
79,8	14,2	5,3	0,7	0	0	0	0	11.FFT
94,9	4,7	0,2	0,2	0	0	0	0	12.LBP
80,3	12	5,7	1,4	0,6	0	0	0	13.GLGLM
71	19,6	8,6	0,5	0,3	0	0	0	14.GLRLM

Table 5.6: Cumulative distribution of the shortest interval lengths to retrieve first relevant

Observing Tables 5.6, 5.4 and 5.2, we see that the first relevant is surely in the first 100 out of 1000 images. Combinatory algorithms may reduce this search space further. Cooccurence based algorithms including CM, GLVM, NGLDM, GLGLM and GLRLM have problems because they lack of transformational invariants. Nevertheless, since the variants of a requested image are usually closer in distance to each other than the other samples, we observe medium level success rates.

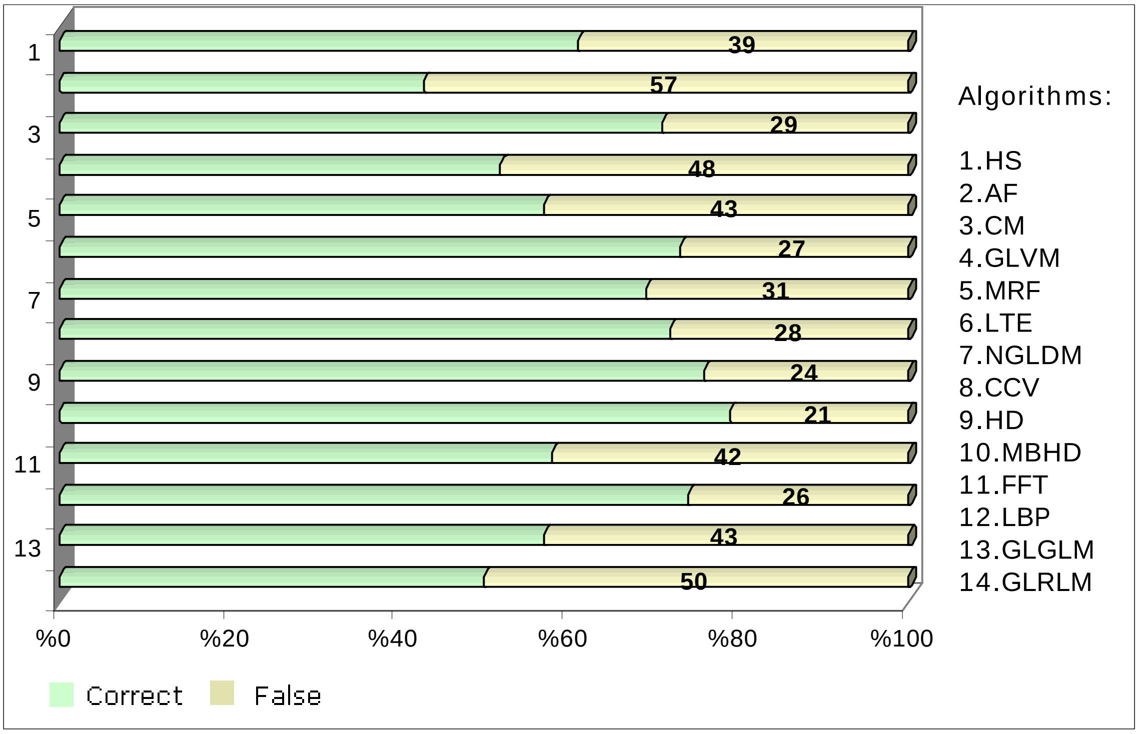


Figure 5.11: The cumulative correctness for the closest retrieval

In Figure 5.11, we observe that the retrieval performances are not much promising if just the closest distance metric is taken into account. This shows that the problem at hand is different than what we call as sameness analysis or classification problem. The aim in content-based retrieval is not to find the requested image as the first guess but reduce the search space.

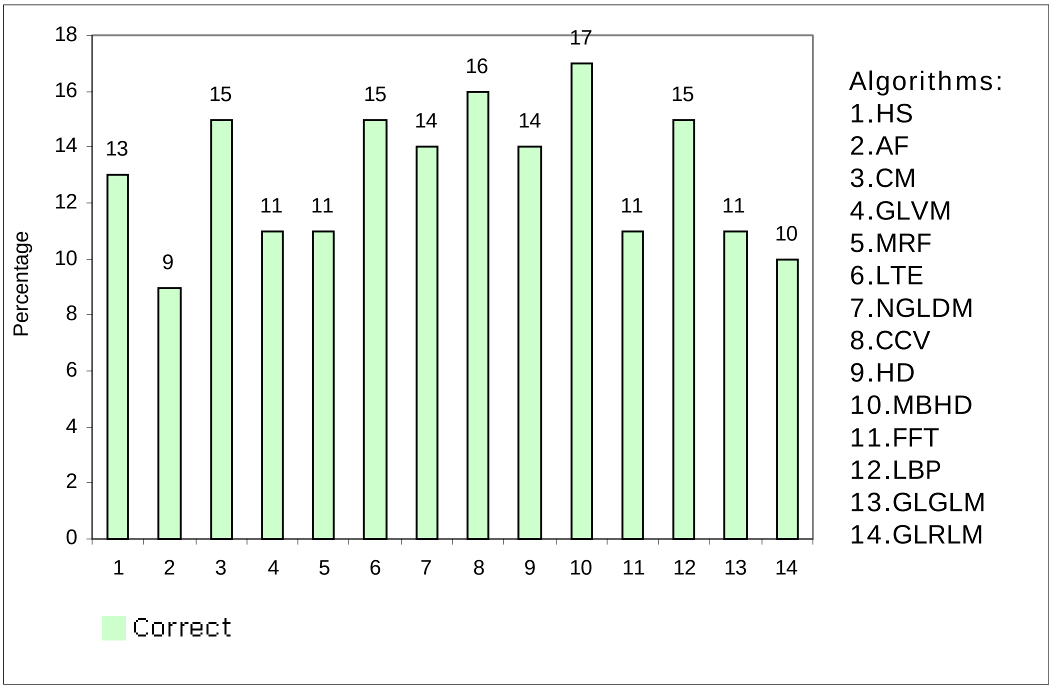


Figure 5.12: The distribution of the first three all correct retrievals

Figures 5.8 and 5.12 are very closely related. It shows that the feature space of the Brodatz database is distributed quite homogenous and the optimum retrieval cases don’t represent the power of content based algorithms.

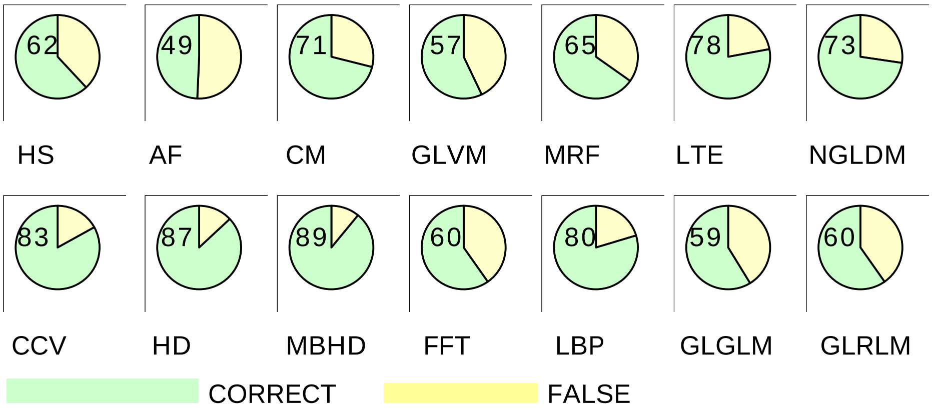


Figure 5.13: The percentage of the first three correct retrievals –majority rule applied

As the number of samples increases, we see in Figure 5.13 that the performances drop with respect to Figure 5.5 and 5.9. Concluding our comments, the MBHD seems to be a healthier metric of colour information; CCV, LBP and LTE are the secondary superior algorithms should be utilised in semantic solutions for texture metrics. CM, GLVM, NGLDM, GLGLM, GLRLM and MRF methods didn’t perform as successful as they are in defect analysis. This is probably because they are affected from small geometric transformations. For a further research, the successful algorithms may be combined to obtain a higher dimensional metric space invariant to the transformations.

CHAPTER 6

CONCLUSION AND FUTURE WORK

Texture and colour algorithms used in defect analysis aim to model a non-defective training sample well and compare the model's sameness with the test images.

The 2D puzzle problem for the archaeology is of the extreme opposite side. The continuation of two images is required to be found to define them as neighbours.

To ease the problem and prepare the required background knowledge of texture, we have worked exactly on the midway, on the similarity analysis. The content based image retrieval is an excellent performance benchmark place to extract the basics of successful similarity analysis.

On the basis of feature performances, we reached quite promising results in comparison to the other methods stated in the literature[1,4,10]. This may be because we combine many robust algorithms of the machine vision, pattern analysis and image processing on which we have experiences.

But, we skipped a very important variable for the comparison of different content-based systems, the computation time. This is because of the problem type at hand:

As a final goal, we would like to solve the archaeological puzzle problem, which will minimise the time from the several years, if done manually, to a couple of weeks with a computerised method. Therefore, we don't need to bother with the computation complexity much, although we've tried to write the codes as if we do the job for an image-based content system.

Another important point is that the results shouldn't be compared with the classification performances of the same algorithms. As stated earlier, the methods are specially designed for the machine vision to group textures into two classes, defective and non-defective. Besides, content retrieval studied and available in the literature is categorised as level 1 or low level, which needs at least other semantic associations to mimic a successful retriever like the human eye. In biological vision, we never compare instantly 1000 images and report as a 1000x1000 matrix. But in fact, all these synthetic obstacles helped us to decide what kind of a system we face and where we are in knowledge space.

The future of the study is to bring the whole content based system to a literally optimal condition. The pre-processing algorithms, the pattern analysis techniques should be made more robust, different feature extraction techniques should be employed together and further comparisons should be made to catch the works on multimedia mining started as early as 1980s.

The puzzle problem using textural features is original and before solving the continuity problem, the similarity analysis should be well understood. This thesis is a first step in building this knowledge. But more is needed:

Global features of the pieces are as important as the ones that we will track in the neighbourhood zone. That means, we should account for global object properties as well as continuous piecewise features. These are what we call semantic features.

The sizes and boundaries of the images are transformed to an uncontrolled space which requires advanced string matching algorithms. Because of this freedom, the global features should be scale and rotation invariant but we need to use the direction and area feature differentiation locally.

Memory functionality is needed to mimic the human vision system. For the occlusion cases we plan some guess type of behaviours. Clustering robustness and efficient indexing similar to the one seen in the brain are needed.

And finally, the real archaeological problems are usually in 3D, therefore 2D solutions should be transformed into three dimensional space.

Content based retrieval and the puzzle solving combine many branches of computer science, pattern analysis and image processing together. For a historical country like Turkey, this type of work is valuable for the archaeologists and archivists.

As an initial attempt for texture based projects in our university, this thesis aimed to import the knowledge available in the literature and hopes to trigger ideas for the future.

APPENDIX A

PUZZLE PROBLEM

In archeology, art restoration and failure analysis, we may encounter a large number of irregular fragments resulting from one or several broken objects.

The goal of the computer assisted reconstruction is to assist with the reconstruction of archeological artifacts, such as pottery fragments, marble relief pieces, or mosaics, using computational tools and to electronically disseminate the reconstructed artifacts, through a virtual museum or virtual tours.

Traditionally, the archeological findings are usually labeled by a human expert. As the number of artifacts increase, the task of assembling the pieces together becomes almost intractable, due to combinatorial explosion of all the possible ways the pieces may fit together.

Figure A.1: Hasankeyf archaeological site, Turkey

Automation, precision of the data acquisition and reveal of hard-to-see associations by multimedia mining will be very helpful for the archeologists.

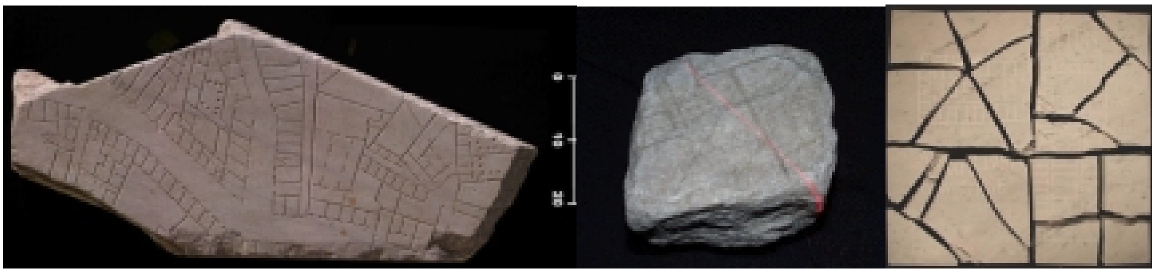


Figure A.2: Digital Forma Urbis Romae Project, Italy

Computational tools to help with this process would be very valuable in speeding up the reconstruction. Using 2D and 3D shape matching techniques, as well as suitable texture similarity analysis, the computer may eliminate pieces that do not match, greatly reducing the number of possible alternatives to search, either by the computer or by humans, which is tedious and laborious.

Current works in the literature specialize on shape matching[61], by which they approach the automatic 2-D and 3-D jigsaw puzzle solving problem in two stages, local shape matching followed by global search and reconstruction. Local shape matching aims to find candidate pairs of matching fragments which are adjacent in the original objects using only local shape information. In finding a global solution, ambiguities resulting from local shape matching are resolved and the pieces are merged together. But as the pieces of the puzzle increase, the system should evaluate a lot of combinations.

Other features can be useful in helping to reduce the search space. We focus on textural similarity, 3D solutions of which are not available on the literature. Therefore, before the final solution for the 3-D object puzzle problem, 2-D works on mosaics, ancient tablet typography solving and 2D-map reconstruction like the one in digital forma urbis romae project should be studied.

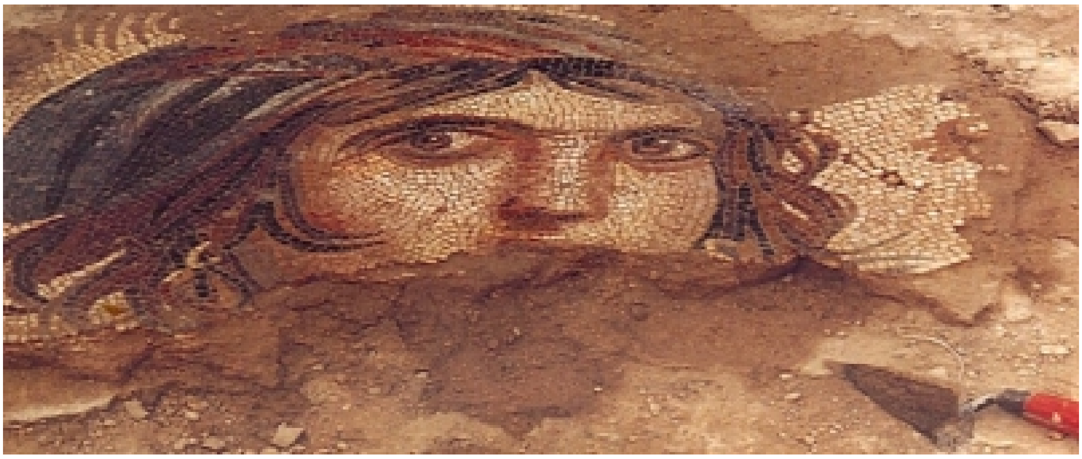


Figure A.3: Zeugma Archaeological Site, Turkey

Once reconstructed, capability of browsing the archeological artifacts through an interactive multimedia environment would be invaluable in preserving and sharing the cultural heritage. Integration of the data will fuse the expert knowledge and will broaden the overall archaeological base.

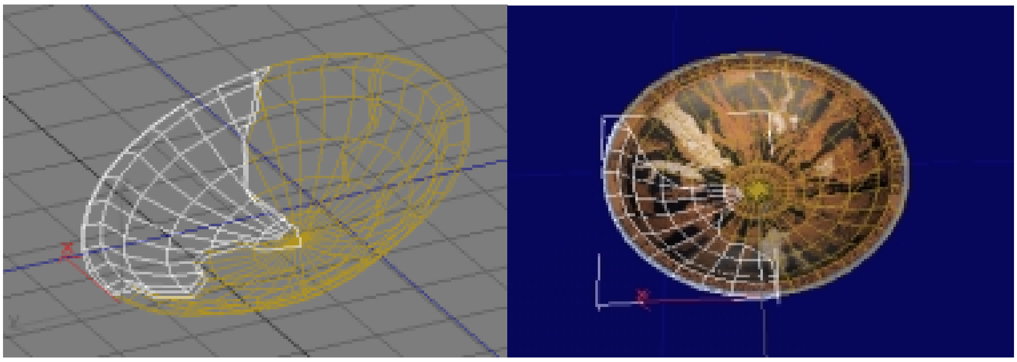


Figure A.4: Computerized cultural heritage

As recent attempts, Stanford University has tried to reconstruct the Severan Marble Plan of Rome on Digital Forma Urbis Project. At the University of Athens, a system called the Virtual Archaeologist has been developed to match 3D sherds modeled via surface patches; Brown University is studying the problem of matching 3D curves taken to be breakcurves of sherds. An international team, led by Brunel University and with support from the European Union, is developing and using 3D multimedia tools to measure, reconstruct and visualize archaeological ruins in virtual reality using as a test case the ancient city of Sagalassos in Turkey.

APPENDIX B

DATABASE SAMPLES

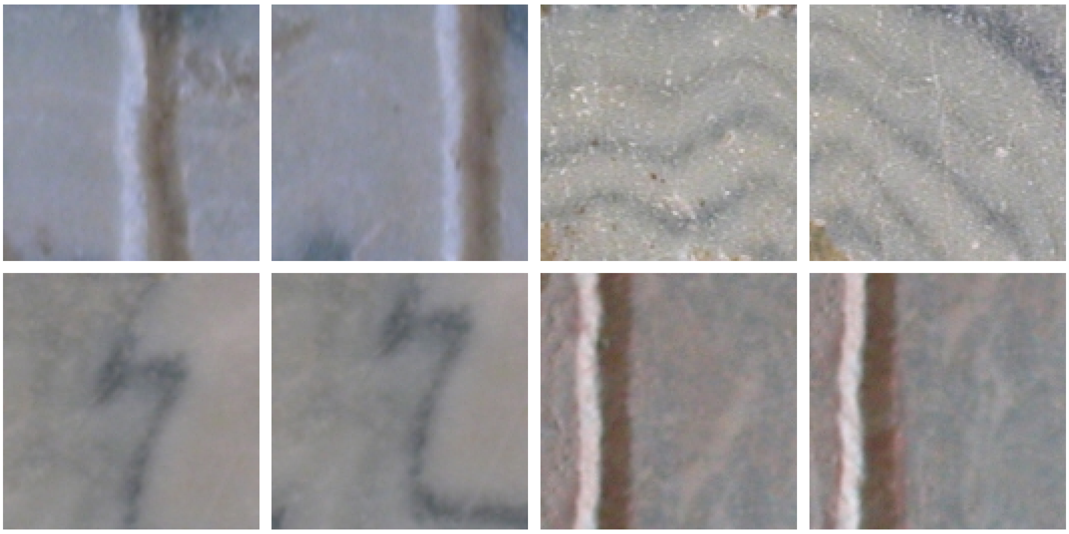


Figure B.1: Samples from the Archaeological Marbles Database

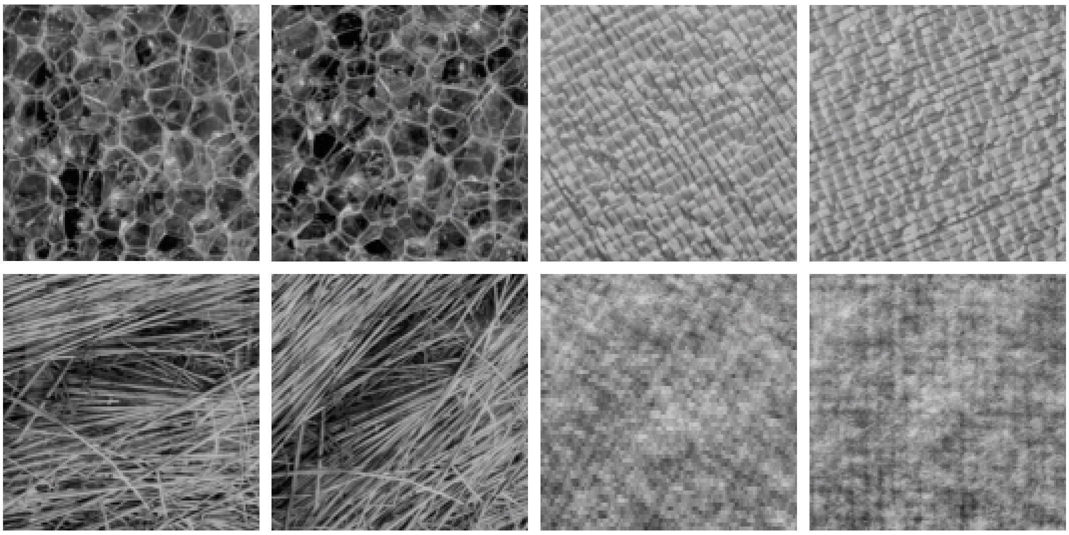


Figure B.2: Samples from the Brodatz Database

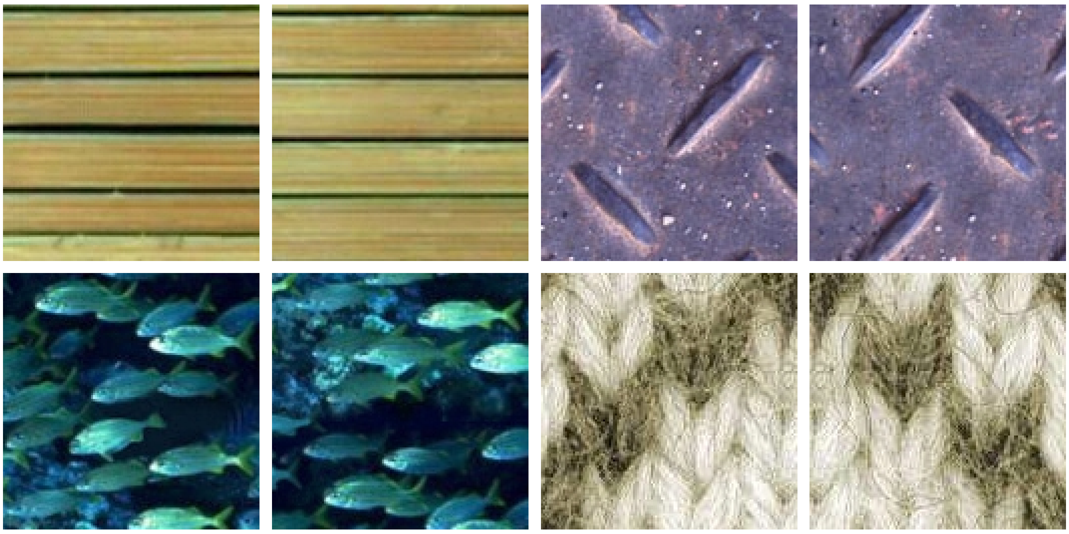


Figure B.3: Samples from the Big Database

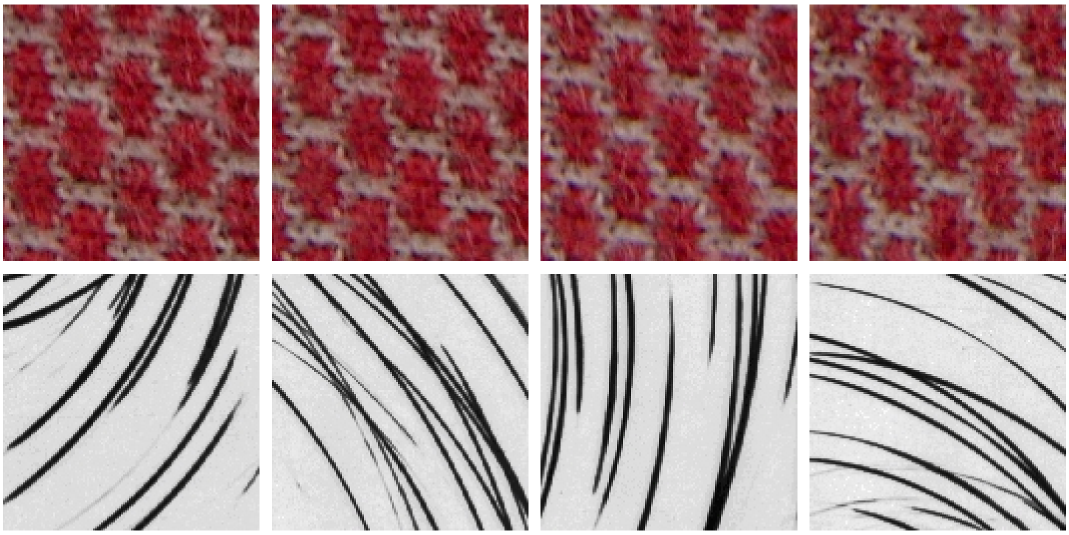


Figure B.4: Typical image classes

REFERENCES

[1] L. G. Shapiro, G. C. Stockman, *Computer Vision*. Prentice Hall Inc., 2001.

[2] E. R. Davies, *Machine Vision*. Academic Press, 1997.

[3] M. Sonka, V. Hlavac, R. Boyle, *Image Processing, Analysis and Machine Vision*. Cambridge University Press, 1993.

[4] A. D. Bimbo, M. Kaufmann, *Visual Information Retrieval*. Morgan Kaufmann Publishers, 1999.

[5] R. C. Gonzales, R. E. Woods, *Digital Image Processing*. Addison-Wesley Publishing Company, 2002.

[6] I. Glynn, *An Anatomy of Thought*. Phoenix, 2000.

[7] H. Barlow, C. Blakemore, M. Weston, *Images and Understanding*. Cambridge University Press, 1990.

[8] M. Seul, L. O'Gorman, M. J. Sammon, *Practical Algorithms for Image Analysis*. Cambridge University Press, 2000.

[9] R. Schalkoff, *Pattern Recognition*. John Wiley & Sons Inc., 1992.

[10] A. Smeulders, R. Jain, *Image Databases and Multi-Media Search*. World Scientific, 1997.

[11] A. K. Jain, *Fundamentals of Digital Image Processing*. Prentice Hall Inc., 1989.

[12] H. M. Deitel, P. J. Deitel, T. R. Nieto, *Internet&World Wide Web-How to Program-*. Prentice Hall Inc., 2000.

[13] S. J. Sangwine, R. E. N. Horne, *The Colour Image Processing Handbook*. Chapman&Hall, 1998.

[14] Robert J. Schalkoff, *Digital Image Processing and Computer Vision*. John Wiley& Sons Inc., 1989.

[15] Bernd Jaehne, *Digital Image Processing*. Springer Verlag, 1995.

[16] Adrian Low, *Introductory Computer Vision and Image Processing*. McGraw-Hill Book Company, 1991.

[17] R. Carter, *Mapping the Mind*. Phoenix, 2000.

[18] S. Pinker, *How the Mind Works*. W.W. Norton & Company Inc., 1997.

[19] K. Koffka, *Principles of Gestalt Psychology*, Harcourt, Bruce and Company, 1935.

[20] A. Atalay, “Automated defect inspection of textile fabrics using machine vision techniques,” *MS thesis, Bogazici University*, Istanbul, 1995.

[21] C. Unsalan, “Pattern recognition methods for textural analysis case study: steel surface classification,” *MS thesis, Bogazici University*, Istanbul, 1998.

[22] G. Pass, R. Zabih, J. Miller, “Comparing images using color coherence vectors,” *In Proceedings of ACM Multimedia 96*, pp. 65-73, Boston MA USA, 1996.

[23] D. M. MacKay, “Strife over visual cortical function,” *Macmillan Journals Ltd.*, 1981.

[24] D. A. Pollen, S. F. Ronner, “Visual cortical neurons as localized spatial frequency filters,” *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 13, no. 5, 1983.

[25] E. L. Thomas, “Movements of the eye,” *Scientific American*, 1968.

[26] H. Tamura, S. Mori, T. Yamawaki, “Textural features corresponding to visual perception,” *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 8, no. 6, 1978.

[27] A. Kankanhalli, H. J. Zhang, C. Y. Low, “Using texture for image retrieval,” *International Conference on Automation, Robotics and Computer Vision*, Nov. 1994.

[28] T. Ojala, M. Pietikaenen, D. Harwood, “Performance evaluation of texture measures with classification based on kullback discrimination of distributions,” *IEEE*, 1994.

[29] T. Ojala, M. Pietikaenen, J. Nisula, J. Heikkinen, “Experiments with two industrial problems using texture classification based on feature distributions,” *Intelligent Robots and Computer Vision*, 1994.

[30] R. W. Picard, T. P. Minka, “Vision texture for annotation,” *Multimedia Systems*, vol. 3, 1995.

[31] P. P. Ohanian, R. C. Dubes, “Performance evaluation for four classes of textural features,” *Pattern Recognition*, vol. 25, no. 8, 1992.

[32] R. Muzzolini, Y. Yang, R. Pierson, “Texture characterization using robust statistics,” *Pattern Recognition*, vol. 27, no.1, 1994.

[33] B. S. Runnacles, M. S. Nixon, “Texture extraction and segmentation via statistical geometric features,” *IEEE*, 1996.

[34] T. Ojala, M. Pietikainen, “Multiresolution gray-scale and rotation invariant texture classification with local binary patterns,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 24, no. 7, July 2002.

[35] D. He, L. Wang, “Texture features based on texture spectrum,” *Pattern Recognition*, vol. 24, no. 5, 1991.

[36] J. Strand, T. Taxt, “Local frequency features for texture classification,” *Pattern Recognition*, vol. 27, no. 10, 1994.

[37] L. Van Gool, P. Dewaelle and A. Oosterlinck, “Survey-texture analysis anno,” *Computer Vision, Graphics and Image Processing*, vol. 29, pp.336-357, 1985.

[38] C. H. Chen, L. F. Pau, P. S. P. Wang, *The Handbook of Pattern Recognition and Computer Vision*. World Scientific Publishing Co., 1992.

[39] P. P. Ohanian, R. C. Dubes, “Performance evaluation for four classes of textural features,” *Pattern Recognition*, vol. 25, no. 8, 1992.

[40] C. Unsalan, A. Ercil, “Classification of rust grades on steel surfaces part 1&2&3,” *FBE-IE-12/97-16, Bogazici University*, 1997.

[41] A. Zalesny, L. V. Gool, “Multiview texture models,” *IEEE*, 2001.

[42] M. Hauta-Kasari, J. Parkinen, T. Jaaskelainen, R.Lenz, “Generalized coocurance matrix for multispectral texture analysis,” *13th International Conference on Pattern Recognition*, ICPR’96.

[43] L. S. Davis, S. A. Johns, J. K. Aggarwal, “Texture analysis using generalized co-occurance matrices,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. PAMI-I, no. 2, 1979.

[44] C. R. Dyer, T. H. Hong, A. Rosenfeld, “Texture classification using gray level coocurence based on edge maxima,” *IEEE Transactions on Systems, Man and Cybernetics*, vol. SMC-10, no. 3, 1980.

[45] A. Latif-Amet, A. Ertuzun, A. Ercil, “An efficient method for texture defect detection:sub-band domain co-occurance matrices,” *Image and Vision Computing*, 1999.

[46] V. Kovalev, M. Petrou, “Multidimensional co-ocurence matrices for object recognition and matrices,” *Graphical Models and Image Processing*, vol. 58 no. 3, 1996.

[47] R. Chellappa, S. Chatterjee, “Classification of textures using gaussian markov random fields,” *IEEE Transactions on Acoustics, Speech and Signal Processing*, vol. ASSP-33, no. 4, 1985.

[48] Z. Kato, M. Berthod, J. Zerubia, “A hierarchical markov random field model and multitemperature annealing for parallel image classification,” *Graphical Models and Image processing*, Jan. 1996.

[49] S. Ozdemir, A. Ercil, “Markov random fields and karhunen-loeve transforms for defect inspection of textile products,” *Proceedings of IEEE ETFA'96*, vol. 3, pp. 697-703, Hawaii, 1996.

[50] F. S. Cohen, Z. Fan, S. Attali, “Automated inspection of textile fabrics using textural models,” *IEEE Transactions on Pattern Analysis and Machine Vision Intelligence*, vol. 13, no. 8, 1991.

[51] I. M. Elfadel, R. W. Picard, “Gibbs random fields, coocurances and texture modelling,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 16, no. 8, 1994.

[52] B. S. Manjunath, R. Chellapa, “Unsupervised texture segmentation using markov random field models,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 13, no. 5, 1991.

[53] G. R. Cross, A. K. Jain, “Markov random field texture models,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. PAMI-5, no. 1, 1983.

[54] W. Xinli, F. Albregtsen, B. Foyn, “Gray level gap length matrix: a new approach for texture analysis,” *The Third International Conference on Automation, Robotics and Computer Vision*, Nov. 1994.

[55] J. Eakins, M. Graham, “Content-based image retrieval,” *A report to the JISC Technology Applications Programme*, University of Northumbria at Newcastle, Jan. 1999.

[56] C. C. Venters , M.Cooper , “A review of content-based image retrieval systems,” *JISC Technology Applications Programme*, University of Manchester, 1999.

[57] A. L. Amet, “Texture defect detection using wavelet transforms,” *MS thesis, Bogazici University*, Istanbul, 1997.

[60] G. Gagaudakis, P. L. Rosin, “Incorporating shape into histograms for CBIR,” *Pattern Recognition*, 2002.

[61] W. Kong, “On solving 2D and 3D puzzles using curve matching,” *MS thesis, Brown University*, May 2002.

[62] J. P. Eakins, “Towards intelligent image retrieval,” *Pattern Recognition*, 2002.

[63] W. Xinli, “Gray Level Gap Length Matrix: A New Approach for Texture Analysis,” *Proceedings of the Third International Conference on Automation,Robotics and Computer Vision(ICARCV’94)*, Nov. 1994.

[64] K. Yogesan, “Gray Level Variance Matrix: A New Approach to Higher Order Statistical Texture Analysis,” *Proceedings of the Third International Conference on Automation,Robotics and Computer Vision(ICARCV’94)*, Nov. 1994.