UNIVERSIDAD DE LAS PALMAS DE GRAN CANARIA

DEPARTAMENTOS DE INFORMÁTICA Y SISTEMAS

TESIS DOCTORAL

MODELOS DE PERCEPCIÓN VISUAL BASADOS EN LA ORIENTACIÓN DE CONTORNOS

MIGUEL ALEMÁN FLORES

Las Palmas de Gran Canaria, Enero de 2002

53/2001-02

UNIVERSIDAD DE LAS PALMAS DE GRAN CANARIA
UNIDAD DE TERCER CICLO Y POSTGRADO

Reunido el día de la fecha, el Tribunal nombrado por el Excmo. Sr. Rector Magfco. de esta Universidad, el/a aspirante expuso esta TESIS DOCTORAL.

Terminada la lectura y contestadas por el/a Doctorando/a las objeciones formuladas por los señores miembros del Tribunal, éste calificó dicho trabajo con la nota de Sobresaliente "cum laude"

Las Palmas de Gran Canaria, a 15 de marzo de 2002.

El/a Presidente/a: Dr.D. Roberto Moreno Díaz,

El/a Secretario/a: Dra.Dña. Olga María Bolivar Toledo,

El/a Vocal: Dr.D. Nicholas Leibovic,

El/a Vocal: Dr.D. Dario Maravall Gómez-Allende,

El/a Vocal: Dr.D. Christopf Schnorr,

El Doctorando: D. Miguel Alemán Flores,

Universidad de Las Palmas de Gran Canaria
Departamento de Informática y Sistemas
Programa de Percepción Artificial y Aplicaciones

TESIS DOCTORAL

Modelos de Percepción Visual basados en la Orientación de Contornos

MIGUEL ALEMÁN FLORES

Las Palmas de Gran Canaria
Enero de 2002

Universidad de Las Palmas de Gran Canaria
Departamento de Informática y Sistemas
Programa de Percepción Artificial y Aplicaciones

Modelos de Percepción Visual basados en la Orientación de Contornos

Tesis Doctoral presentada por
D. Miguel Alemán Flores

Dirigida por los Drs.
D. Roberto Moreno Díaz (hijo) y D. Luis Álvarez León

Las Palmas de Gran Canaria
Enero de 2002

El doctorando | *Los directores*

Miguel Alemán Flores | Roberto Moreno Díaz (hijo) | Luis Álvarez León

Quisiera dedicar este trabajo a mis padres y a mis hermanos.

También quiero agradecer el apoyo y los consejos de mis compañeros y amigos Laura Cruz, Francisca Quintana, Abraham Rodríguez y Agustín Trujillo.

A Roberto Moreno hijo, por haberme introducido en el mundo de la visión artificial, guiando mis primeros pasos en el terreno de la investigación.

A Luis Álvarez, por su dedicación y sus continuas explicaciones, sugerencias y correcciones, con las que me ha transmitido conocimientos y formas de trabajo.

Y a todos aquéllos que de modos diferentes han contribuido a la realización de esta tesis.

Modelos de Percepción Visual basados en la Orientación de Contornos

Visual Perception Models based on Contour Orientation

Contents

Chapter 1

INTRODUCTION

The sense of vision is, in many species, the one that provides the largest amount of information about the milieu of the individual, thus being that which most influences on its behavior. This information can be used in many different ways according to the conditions, needs and situation of the animal or system under consideration. In order to obtain food, to escape out of the sight of a predator, to follow the individuals of the species or any other vital function, the location and identification of the objects of the environment, the determination of their sizes and the extraction of their trajectories are essential aspects of a visual system, in either a biological or man-made environment. Even more, sometimes it is also necessary to determine the nature of the material the objects are made of.

In this work, we present a model for estimating these parameters by using a set of filters and primitives which simulate the activity of the receptive fields in a higher vertebrate retina. Even if the mechanisms which have been used to build these structures and exploit the possibilities to use the information they provide for many higher level processes are far from the real architecture of the natural systems, they endow us with a strong inspiration in the search for new techniques. These parameters were implicit in the classical description of the frog's retinal ganglion cells by Lettvin et al. [LMMP59], and since the work of Barlow et al. [BHL64] on speed sensitive cells in the rabbit, a substantial literature has grown up concerned with both, the biological and the computational problems.

1.1 General description

Due to the importance of vision in the interaction of animals with the environment, a large proportion of human effort to build machines behaving as living beings has been devoted to artificial vision. Trying to collaborate in this working field, we have developed a framework for the processing of visual information in several channels which are specialized in different aspects of information analysis, such as shape, motion, texture or color, but which preserve certain cohesion elements and make use of the same kind of basic tools.

The identification of an object is provided, in most cases, by a description of its shape, which requires an accurate location of its borders. However, the output of the mechanism used for the recognition should not be altered by changes in the orientation, size or contrast between the object and the background. This work presents a multichannel description of visual information processing in which, from common initial stages where low-level features are extracted, different subsystems analyze the information for more specific purposes, such as shape, motion, texture or color discrimination.

We introduce a set of formal tools, based on Newton filters [Mor93], which allow estimating edge orientation and whose output does not vary when the input signal is rotated or when a global illumination change occurs. Furthermore, the operations are performed in a layered structure which simulates the activity of ganglion cells in higher vertebrate retina. The outputs of these filters are used for several purposes, since the selective detection of the borders is a basis for many other primitives of higher semantic level.

Firstly, it allows us to extract a one-dimensional representation of the contour in order to characterize shapes by using Fourier coefficients. From these coefficients, an energy function is built to discriminate shapes. This work presents a robust method for the characterization of objects which allows discriminating among quite similar shapes, such as those of keys and fishes. The use of an orientation function to represent a shape and its analysis from the Fourier coefficients make it possible to extract relationships for the different ways in which a certain shape may be presented, thus constituting a very useful mechanism for many different purposes in areas like industry, medicine, meteorology, etc.

Secondly, the parameters used when comparing object contours allow studying a sequence of frames and extracting motion information, which represents the introduction of temporal evolution in the scene. The information obtained to relate shapes is also used for a temporal association of frames in which a given object evolves. In the same way as the representation of shapes is based on the extraction of edge orientation in every point on the contour, the process of motion analysis is based on the representation of shapes.

Finally, in the same way as orientation is used to compare contours by means of a frequency analysis of the functions which characterize them, textures can be studied and classified. An orientation histogram embedded in a multiscale framework is built for each texture to describe it and compare it with other patterns. In this case, a musltiscale analysis of the textures provides much more information, since it is in the evolution of the pattern along the set of scales where the differences between the textures can be detected.

The filters used to estimate edge orientation constitute a simple but effective tool for this goal and, at the same time, they are suitable for layered structures such as those found in the natural systems. On the other hand, the Fourier analysis of the orientation functions allows extracting some general features of the shape which is represented. Moreover, we propose to use weighting functions for the coefficients of the different frequencies in such a way that those coefficients whose information is more relevant have a higher weight in the resulting scheme. We analyze the shapes of such weighting functions in order to improve the discrimination and reduce the failure probability.

The numerical experiences are very promising. In particular, we can even discriminate between shapes which are very similar from a perceptual point of view. All these applications of a common basic set of filters endow us with a global framework for visual information processing, where the combination of simple modules produces the abstraction of higher semantic representations of the outside world.

1.2 State of the art

Many works have been published tackling the problems of shape characterization, motion analysis, texture classification and color perception.

Some of them present the modeling of retinal processing, such as [Lei66], [Mor77], [MRR78] and [MR79], and natural structures and procedures have served as models for their simulation.

When dealing with edge extraction and characterization, some sets of filters are commonly used, like those proposed by Sobel, Kirsch, Prewitt and Robinson [SHB99]. Other works, such as Canny's algorithm [SHB99] are concerned with the precise extraction of edges, but they tackle the problem from a different point of view. In this work, we start from Newton filters [Mor93], which have been previously used as neuron-like structures for edge location and description [QAM99].

The work by Zahn and Roskies [ZR72] deals with shape representation using Fourier analysis, but the approach is different in this case since we work with discrete signals and equidistant points. Olson and Huttenlocher [OH97] front the problem with certain variations of Haussdorff's measure. Loncaric presents a survey tackling different shape analysis techniques from various approaches [Lon98].

As said before, we have used Fourier descriptors as a base for shape characterization. They have also been used in different ways and for various purposes, from aircraft to character recognition, by Wallace and Wintz [WW80], Lin and Chellappa [LC87], who also considered partially occluded shapes, and Shridhar and Badreldin [SB84]. Fourier representation allows a wide range of studies of the properties of an object or region, as shown in [KM89]. Other similar representations, such as elliptic Fourier decomposition have also been used, as in [SD92].

Occlusion is differently grappled with in the works by Turney, Mudge and Volz [TMV85], Mokhtarian [Mok97], Bahnu and Ming [BM87] or Kim, Yoon and Sohn [KYS96]. Shashua and Ullman [SU91] describe iterative methods for grouping contours. Eichmann et al. [ELJT90] use Gabor expansions for shape representation.

Motion analysis has been studied in different ways, like the recursive algorithms which include stereo vision described by Yi and Oh [YO97]. The simulation of natural systems in charge of motion analysis include the works by Alemán et al. [ALM97], Miura et al. [MKN95], Prokopowicz and Cooper [PC95], or Pennartz and van de Grind [PG90]. Differential motion analysis is described by Jain et al. [JKS95], Rong et al. [RCC89], and a survey on optical flow can be found in the work by Barron et el. [BFBB92].

Textures have been studied from many different points of view, from energy measures like Laws' measures [Law79] to descriptions like those in the works by Reed et al. [RWW90] or Gotlieb and Kreyszig [GK90].

Multiscale analysis has been previously used for image processing tasks in works like those by Álvarez et al. [AGLM93], Álvarez and Mazorra [AM94] or Lindeberg [Lin94]. In this case, it has been used for texture classification to complement the Fourier analysis on orientation histograms.

A whole survey and comparison of the most relevant works in image processing, including the topics tackled in this study, is available in the work by Sonka et al. [SHB99].

1.3 Main contribution

In this work we propose a wide range of applications of a new set of filters, the modified Newton filters, in different fields of image processing. Using Fourier analysis, multiscale analysis, error minimization methods and many other mathematical tools, and starting at a very accurate estimation of edge orientation, features such as the shape of an object's silhouette, the trajectory of a moving object or the pattern of a texture can be examined very accurately and with very satisfactory numerical results.

We have used the same framework and basic units for a series of applications which we describe below:

Accurate estimation of edge orientation:

The modified Newton filters represent a new approach in the development of neuron-like structures. In this case, the initial outputs are combined in such a way that the post-processing of the information they provide generates much more accurate and useful results than those extracted independently. The properties of these filters regarding rotational invariance and non-null weights, the normalization process to make the output invariant against illumination changes, and the interpolation of the discrete outputs generate a more refined orientation extraction.

Shape representation and discrimination:

The estimation of the orientation results in the possibility of using this local information to characterize and discriminate shapes which are visually very similar and whose differences are very slight, even for a human observer.

We use the Fourier transform of orientation functions as basic tool in our analysis. We have introduced energy functions to measure the similarity of the shapes, and these functions have been weighted in order to enhance the characterization strength of this technique by giving more relevance to those Fourier coefficients which are more significant for the purpose of our work. Furthermore, we have tested different types of functions to consider what shape the weighting function should present to facilitate the discrimination as much as possible. We have applied these new techniques to a database consisting of 1000 shapes of marine animals and we have obtained a very satisfactory classification in a quite difficult situation.

We propose to use the information extracted for a segment of an orientation function to relate different parts of a sequence in such a way that we can associate the visible regions of an object when it is not completely accessible. This requires a common analysis of the different segments which have been extracted, since some conditions must be set in order to guarantee that they do belong to the same shape. Hence, the translation, rotation and scaling relationships must be the same for all couples of associated segments.

Motion analysis:

The accuracy of edge orientation estimations makes it possible to enlarge the range of applications to those in which the refinement of these values favors considerably the parameter extraction, such as motion analysis. In this case, we propose a method to identify an object's evolution according to a similarity transformation in which the shape of the object is preserved, i.e. including translation, rotation and scaling. In this case, the contour and the orientation values we have extracted for shape characterization make it possible to adapt a figure to the corresponding contours in different time instants, and the relation between these contours allows determining the temporal evolution of the object.

Multiscale texture representation and classification:

Going further in the study of an object's properties, we have applied these filters for texture classification. Due to the fact that textures may be presented in different scales and this would affect the orientation histograms which are used for their characterization, we propose to include a multiscale analysis in order to be able to process textures regardless of the scale at which they have been acquired.

The combination of modified Newton filters and multiscale analysis represents a new approach in this kind of processes which generates suitable classifications of texture databases.

Finally, we also present the role of color in all these processes since the basis which is established here can be widely enlarged by considering other factors, such as color, three-dimensional images, etc. From a biological inspiration, we have built a system which is able to process the visual information of a scene in different channels. The parallelism of many of the tasks which have been implemented and the interaction between different subsystems make it possible to design an effective structure for such kind of computations. On the other hand, the basic units are the same for various subsystems, such as shape representation, motion analysis and texture classification. The applicability, adaptability and accuracy of these techniques, together with their modularity, make them suitable for the development of vision-guided systems and the simulation of natural systems.

1.4 Structure of this work

Apart from the present chapter in which the general ideas are introduced, this work is structured into eight other chapters as follows:

- In chapter 2, we describe the natural visual pathway and the different stages in the information processing. This is the system whose imitation in searched for and from which some inspiration is obtained. Hence the importance of knowing how it works and how those functions we are interested in are carried out in the natural system.

- Chapter 3 presents the different kinds of transformations which are performed in the visual pathway, classifying and organizing them into a hierarchy. Due to the interdependencies of the different subsystem which take part in the image processing, the transformations can be structured into a set of levels and categories.

- In chapter 4, Newton filters are introduced and their mathematical properties are described. Afterwards, we show how these filters can be adapted to build a new type of filters, the modified Newton filters, which can be used for the estimation of edge orientation. These are the basic units which provide the information for the further analysis which is described in the following chapters. Shape representation, motion analysis and texture classification are based on the outputs of these filters.

- In chapter 5, we describe how we can build a representation of the contour of an object from the outputs of these new filters and how this representation can be analyzed from its Fourier coefficients. It offers a description of the influence of the starting point of a closed contour on those coefficients and an explanation about the way we can deal with inversely obtained contour representations and reflected shapes. It introduces the energy function which is used to measure the similarity between two shapes and decide whether two objects belong to the same shape category or not. Different weighting functions for the frequency factors are tested and compared to improve the discrimination capability of the energy function by regulating the contribution of each one of them to the global result. This chapter also deals with the identification of partially occluded objects. A database clustering example is shown with images of fishes for which the most similar contours are located. This fish database has been kindly provided to us by Professor Farzin Mokhtarian at the Centre for Vision, Speech, and Signal Processing of the University of Surrey [Mok01].

- In chapter 6 the information extracted from the contours is used to fit objects and analyze their motion, obtaining translation, rotation and scaling parameters. This allows focussing the attention on an object and describing its temporal evolution. The extraction of the contours in different frames allows determining the transformation parameters which bring one of the shapes to the other, once they have been identified as the same object according to their contours.

- Chapter 7 shows the application of the same techniques to the analysis of orientation histograms of textures. The Fourier analysis used in the previous chapters is combined with a multiscale analysis in order to achieve a more robust technique for texture classification. Different situations, such as zooming, darkening, lightening or inversion are considered in order to reach a quite general classification method. As done with the shapes in chapter 5, a database clustering is performed, but in this case, it is the textures of the regions that constitutes the classifying factor.

- Chapter 8 deals with the processing of color in a multichromatic system. It grapples with the problem of enhancing color contrast and combining multichromatic information. At the same time, we explain the possibility of adapting the previous techniques for color images in which the information is tripled.

- Finally, in chapter 9, a brief discussion about the previous topics is presented, including the most relevant conclusions extracted from this work and the future trends.

Chapter 2

NATURAL VISUAL SYSTEMS

The understanding of natural systems has been very useful for the development of artificial systems, and also in completely artificial structures it is possible to detect some biologically inspired similarities. Their simulation, either to study their behavior or to extract new techniques, has made it possible to build new mechanisms which improve the performance of man-made equipment, even if the efficiency of the latter is not comparable to that of the natural structures. Since we are working with image processing, the natural system which is relevant for us is the visual system.

We will study the general structure and anatomy of the human eye, the layers that constitute the retina and their functionality, and the higher organs which take part in the processing of visual information, considering first the global description of a neuron. Finally, we will describe how orientation, motion and color are processed in natural visual systems.

2.1 The eye

Higher vertebrates perceive visual information through their eyes. Structurally, the outer ball of the eye consists of three layers, which are choroid, sclerotic and retina. It is in the retina where the sense of vision takes actually place, being the other layers the organs in charge of supplying the necessary support and nutrients to the retina. Before reaching the receptors in the retina, the light must cross the whole eye, entering through the pupil's aperture, being corrected by the crystalline and traversing the different humors which fill the ocular globe. At the end of this way, the photoreceptors, which are indeed the first elements of the retina which capture the light, work as transducers, i.e. convert the light signal into an electrical impulse which can be interpreted and processed by other kinds of neurons in the retina. A general overview is shown in figure 2.1.

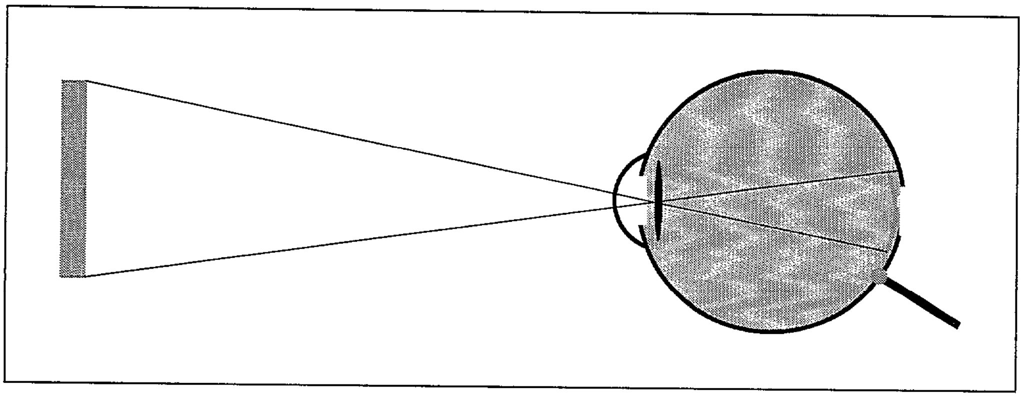


Figure 2.1: Light crosses the cornea, the pupil and the crystalline and reaches the retina, where it is actually acquired.

Due to the fact that the photoreceptors constitute the outer layer of the retina, the signals travel through the inner region in subsequent processing tasks. At the end, the axons of the ganglion cells forming the optic nerve which brings the information to the central organs, must exit the eye. That is the reason why a part of the back region of the eye, the optic disc, does not contain receptors and thus, the light impinging the retina in that area cannot be perceived. Only binocularity makes it possible to compensate this blind point.

The central area of the retina, called fovea, is the region where visual acuity is higher, thus allowing to perceive the objects with a higher resolution as in the peripheral areas. Furthermore, the distribution of the different types of receptors in that region and in the periphery is also unequal and, consequently, the perception capability is affected.

For the combination of both images, when dealing with stereo vision, the temporal (lateral) area of one of them is combined with the nasal (central) area of the other. The areas corresponding to the same side are called ipsilateral and the opposite are called contralateral (see figure 2.2).

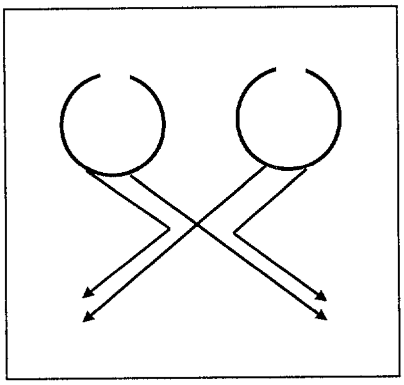


Figure 2.2: Binocular image combination. Ipsilateral regions are transmitted to the same areas in the superior organs.

2.2 Neurons

The organs of a living being are made up of cells. In animals, the cells in charge of the transmission of information across the body are the neurons. A two-way transmission can be discerned: from the outside world, the receptors capture the information to be processed and interpreted; from the central neural system, orders emerge to activate the muscles. As transmitters of information, they are specialized in the reception of signals, their processing and their emission to other neurons which are able to interpret them. In the case of motor neurons, the end of the path is connected to muscles which carry out the order they transport.

For the purpose of this work, sensory neurons are the relevant ones. In this case, the transducers of information are located at the beginning of the path, since the outer signals must be translated into an understandable set of impulses. Typical sensory neurons are characterized by a set of tree-shaped ramifications which are in charge of receiving the impulses which other neurons transmit. This input gates are called dendrites. They carry the impulses to the cell body and after the processing, an output branch, normally longer than the dendrites, is responsible for their conduction to other neurons. This is the axon (see figure 2.3).

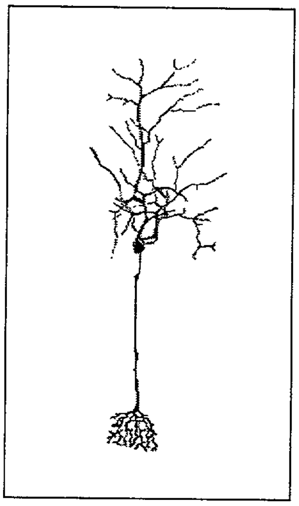


Figure 2.3: Standard neuron structure: dendrites, cell body and axon.

In order to carry the information in the form of these impulses, a sodium/potassium bomb is in charge of creating certain electric potentials which are transmitted along the neuron's membrane. In fact, the potential of the membrane when the neuron is not excited by an impulse is modified by means of the alteration of the interchange of sodium and potassium ions.

By means of certain neurotransmitters, the neurons propagate their impulses to their neighbors in a phenomenon called synapse. A set of terminal bombs and calcium channels make it possible to transmit the information codified in the electrical impulses to the following step.

The cells that constitute the retina are indeed neurons. However, this general description must be adapted since their specialization makes them different in structure and function. Even when comparing the different layers of the retina and the higher organs which take part in visual acquisition and processing, many kinds of neurons appear with different shapes, connections and behaviors. The following section describes the structure and functions of the different neural organs which take part in the acquisition and processing of visual information.

2.3 The visual pathway

The ideas in this chapter are mainly extracted from [Kan81]. The absorption of light and its subsequent translation into electrical signals is carried out by the photoreceptors. The human retina contains two types of photoreceptors, rods and cones. Cones detect form and color, and are responsible for day vision, while rods mediate night vision. They function in the dim light that is present at dusk or in the dark. Under these conditions most stimuli are too weak to excite the cone system. Except in the fovea, the central region of the retina where the acuity is higher, there are 10 times more rods than cones in the retina. Nevertheless, cones are much more important to vision because their loss produces legal blindness, whereas total loss of rods produces only night blindness.

The synaptic events subsequently involved in the transfer of information from the receptor cells to other neurons in the retina have been well documented. Although there are many subclasses of neurons, the vertebrate retina consists of only five major classes: receptor cells, bipolar cells, horizontal cells, amacrine cells and ganglion cells. Figure 2.4 shows an scheme of the retina.

Both types of receptor cells, rods and cones, make direct synaptic contact with a class of interneurons called the bipolar cells, which connect the receptor cells with the ganglion cells. The ganglion cells are the projection neurons of the retina, i.e. they relay visual information to the central nervous system by projecting to the lateral geniculate nucleus and the superior colliculus as well as to brain stem nuclei. Modulating the flow of information from receptor to bipolar to ganglion cells are two classes of neurons: the horizontal cells and the amacrine cells. The horizontal cells mediate lateral interactions between receptor and bipolar cells. The amacrine cells mediate lateral interactions between bipolar cells and ganglion cells.

The cell bodies of these five classes of neurons are found in three layers: the outer nuclear layer (receptor), the inner nuclear layer (bipolar, horizontal and amacrine) and the ganglion cells layer (ganglion). The processes of these five major classes interact in two distinct synaptic layers: the outer plexiform layer contains the processes of receptor, bipolar and horizontal cells; the inner plexiform layer contains the processes of bipolar, amacrine and ganglion cells.

Thus, each ganglion cell reacts to stimuli in a certain area of the retina, its receptive field. The receptive field of an on-center ganglion cell has an excitatory central area and an inhibitory surrounding area. On the contrary, the receptive

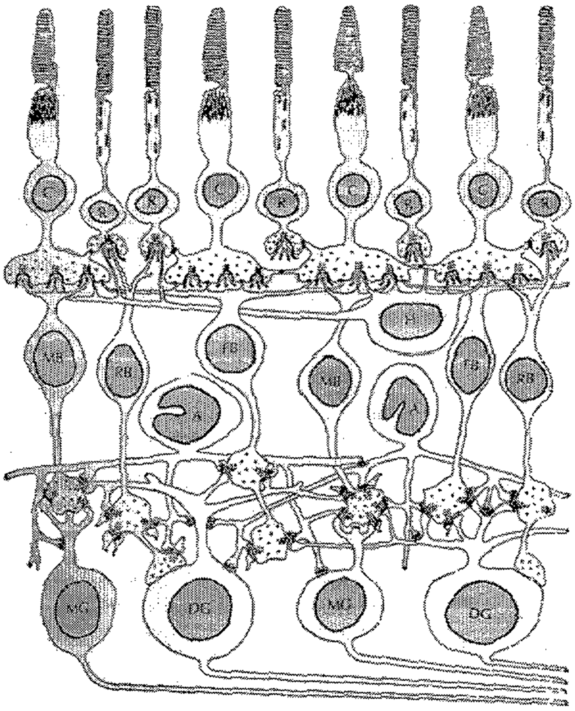


Figure 2.4: Retinal cells: cones (C), rods (R), horizontal cells (H), bipolar cells (MB, RB, FB), amacrine cells (A) and ganglion cells (DG, MG) [Sla90].

field of an off-center ganglion cell has an inhibitory central area and an excitatory surrounding area.

In addition to differences in the location of the dendritic processes of on- and off-center ganglion cells, each retinal region has several morphological and functionally distinct subsets of ganglion cells that subserve the same photoreceptors in parallel. The X cells have medium-size cell bodies and small dendritic fields and participate in high-acuity vision. The Y cells have the largest cell bodies, a large dendritic arborization and rapidly conducting axons. The Y cells respond only to large targets and are important in the initial analysis of crude form. The W cells have small cell bodies and large dendritic arborizations. These cells project to the superior colliculus and are involved in head and eye movement.

The flow of information from receptor to ganglion cells can be best followed by considering two major pathways available in the retina. The first is the simplest and is a direct route from receptor to bipolar to ganglion cell. This pathway carries information from nearby receptors to ganglion cells. In the second pathway, the surrounding horizontal cells integrate and transfer information from distant receptor cells to the bipolar-ganglion cells pathway. The retina is simpler than the brain because it uses only five basic neuronal types. Nonetheless, it can generate complicated properties that reflect considerable transformation of visual information.

The degree of specificity in the central connections of the visual system is remarkable. Separate regions in the retina project upon the lateral geniculate nucleus in such a way that a complete representation of the contralateral visual hemifield is established in the thalamus. Furthermore, distinct cell types occupying the same retinal locus project their axons to different targets in the brain stem, some cells project to the thalamus, some to the midbrain, others to both. The lateral geniculate nucleus is mapped onto the primary visual cortex in a point-to-point manner, since each geniculate axon terminates in and contacts only a small part of layer IV. Cells in layer IV and in the other layers of area 17 have their own highly stereotyped patterns of connections.

The axons of the ganglion cells carry the information to the Lateral Geniculate Nucleus, the Superior Colliculus and the Pretectal Region, but the largest quantity is projected to the first one. In the Lateral Geniculate Nucleus there is an ordered representation of the visual field. However, as in the visual cortex, not all parts of the retina are equally represented. A large part of the Lateral Geniculate Nucleus is devoted to the central region. Their receptive fields are similar to those of the ganglion cells, but the antagonism center-periphery is even stronger.

2.4 Visual processing in natural systems

Due to the inspiration that natural systems have provided in the development of artificial mechanism in visual processing, in this section we study how the processing of information takes place in the natural visual pathway previously described.

2.4.1 Sensitivity to edge orientation in natural systems

The first task we will deal with is the detection of the contrast between neighboring areas in order to extract the borders. Recordings of the activity of single ganglion cells show they are never silent, even in the dark, but light modulates their spontaneous activity. Each cell responds to light, and the most effective stimulus for each cell is a spot of light directed to a specific area of the retina. This area is called the receptive field of the cell. The receptive field of a single cell in any part of the visual system is that area in the retina where stimulation with light causes either excitation or inhibition of the cell's firing pattern [Kan81]. Figure 2.5 shows the structure of center-surround receptive fileds.

Cells of the retina and lateral geniculate fall into two classes: on-center and off-center. Neurons of the primary visual cortex also fall into two major classes: simple and complex. Each of these classes, moreover, has several subclasses. All simple cells are characterized by three features: specific retinal position, discrete excitatory and inhibitory zones and specific axis of orientation [HW62].

The receptive field of a simple cell in the primary visual cortex has clearly delineated excitatory and inhibitory zones with a specific axis of orientation. The receptive field has a narrow central excitatory area flanked by symmetrical inhibitory areas. The best stimulus for this cell is a bar in the center of its receptive field with the appropriate orientation.

The receptive field of a complex cell in the primary visual cortex has no clearly defined excitatory and inhibitory zones. The cell responds best to an edge with a certain orientation. Orientation is important but position within the field is not critical. It is the combination of the outputs of a group of cells in a certain level that makes it possible to extract more relevant information in a subsequent level.

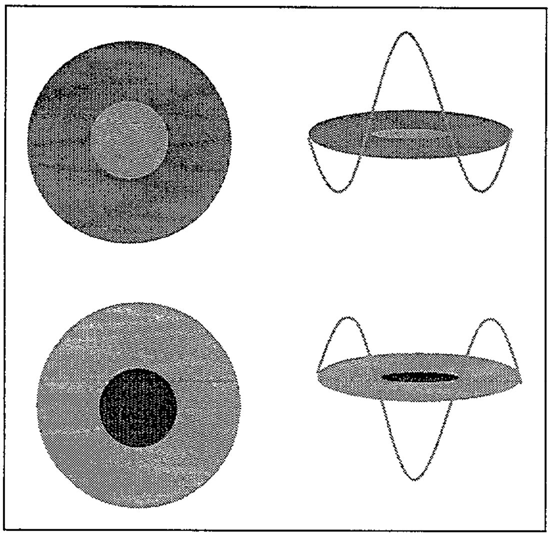


Figure 2.5: Center-surround receptive fields.

2.4.2 Motion sensitivity in natural systems

The researches carried out by Barlow, Hill and Levick [BHL64] showed the way ganglion cells of the rabbit work, and according to their results, they could be classified into 4 main groups. Some of them (48%) had concentric ON or OFF center receptive fields, with inhibitory surrounding. Excluding some cells that were difficult to classify, the rest of the ganglion cells which were studied (41%) showed a response that was much higher when the stimulus moved in a certain direction, called preferred direction, than when it moved in any other direction. Besides, 75% of them reacted to ON as well as OFF stimuli, whereas 25% reacted only to ON stimuli. This preferred direction varies among the cells, thus providing a whole range of directions to be detected. The size of their receptive fields also varies proportionally to the speed of motion, with small receptive fields for cells which detect slow movements and large receptive fields for cells which detect fast movements. Figure 2.6 shows the structure of motion detection cells.

The response of an isolated cell is maximum for the preferred direction, minimum for the null direction, i.e. the direction opposite to the preferred one, and medium for the intermediate directions. The speed to which the cells react are also selected, with a range within which there is a significant response, and out of which the response is not considered as important. If the movement is fast, the response is brief and there is no response to the null direction. But if the movement is slow, there is also a response to the null direction. It is also important to say that the cells react faster to the preferred direction and the response is longer. These direction selective cells are strongly inhibited by the surrounding region in such a way that they are very effective to detect the motion of small stimuli.

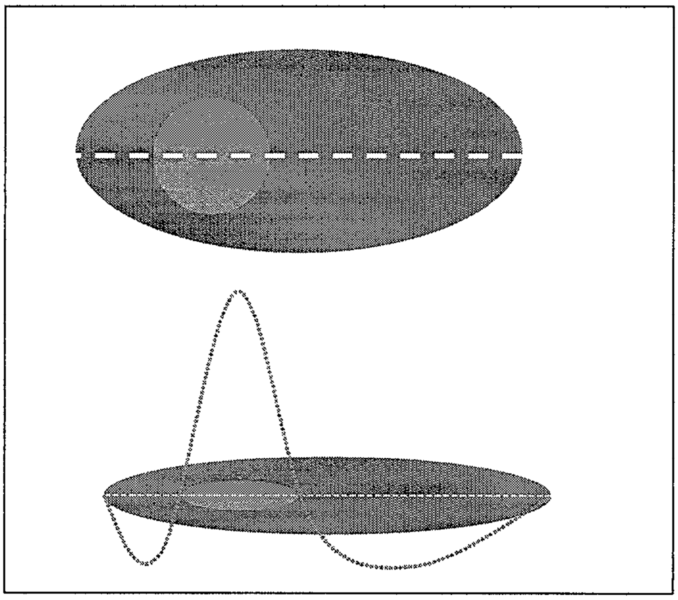


Figure 2.6: Motion detection receptive fields with direction selectivity.

2.4.3 Color perception in natural systems

The color of an object is determined by the wavelength of the light that this object reflects from the exterior light that falls on it, and that reflected light causes certain reactions on our retina to produce the sensation of color. From all possible wavelengths, each species is sensible to a range of them. Our eyes are only sensitive to an interval from 400 to 700 nm. However, that does not mean that we identify them in a linear scale, since it is a three-dimensional scheme (see figure 2.7).

In our retina, we find two types of photoreceptors, a first group specialized in low illumination vision (scotopic vision), the rods, and others mostly specialized in high illumination vision (photopic vision), the cones. Among these latter, there are cells specialized in ranges of wavelength around a certain preferred one which might be grouped into three categories: red, blue and green. As the ranges are overlapped, it is the combination of reactions that determines the real wavelength of the light that will be codified by three parameters. In this way, colors such as yellow, cyan, magenta or even different types of what we call red, green or blue, are combinations of these three categories.

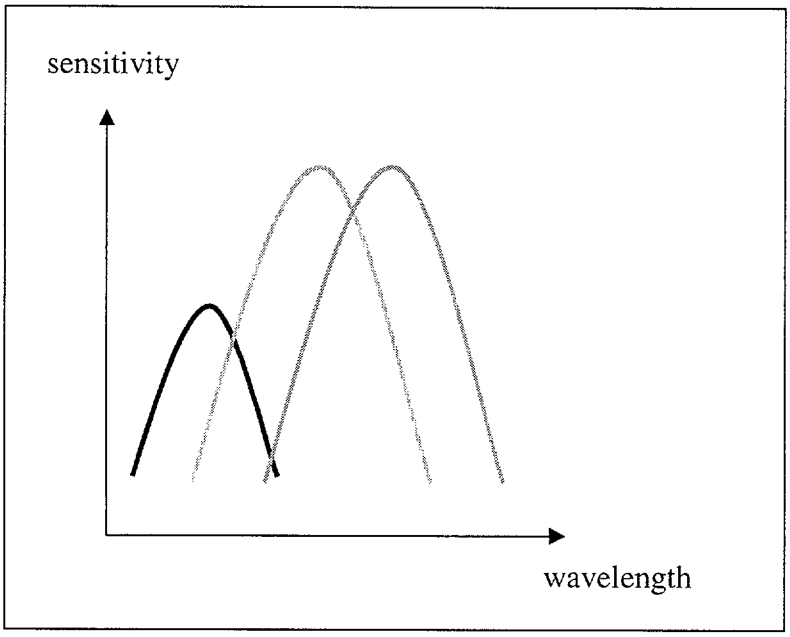


Figure 2.7: Color sensitivity.

We could firstly think that a single type of cones would be enough to identify a range of wavelengths if this type had a broad enough selectivity range. However, the response of the cones is determined not only by the wavelength, but also by the intensity of the light. Besides, the curve of reaction has the same value for different lengths, being almost symmetrical. Consequently, completely different lights would cause the same reaction. If we had two types of cones, the problem would be only partly solved, as some combinations of light could not be distinguished. But since we have three types of cones that cover the range overlapping with the others, it is possible to identify the light by the combination of the three values.

Some species, like pigeons, have four types of cones. Three of them are similar to those in the human retina, but the fourth is devoted to infrared wavelengths, which provides them with a more accurate information for orientation. On the other hand, some species are not sensitive to all the wavelengths we can perceive. For example, the frog's interests determine the perception of colors in such a way that blue is perceived as especially distinct, but not other colors.

Chapter 3

TRANSFORMATIONS IN THE VISUAL PATHWAY

We have already seen how information flows across the different stages of the visual pathway. Even if we have considered it as a whole, we can classify the different items as belonging to a certain category according to the purpose they serve for. So, there are four channels in the retina which transmit information to the Lateral Geniculate Nucleus. Two of them are based on cells with ON center receptive fields and the other two are based on OFF center receptive fields.

For each couple, there are two groups, a first one which transmits rough information with a low resolution and which provides an initial analysis of the scene, Channel Y, and others which process information in a more accurate and refined way, Cannel X. According to the specific function, two channels can be distinguished, one for shape and color, and another for motion and attention.

This chapter introduces a hierarchy in the set of transformations which appear in a visual system. It also considers the consequences of distributed and concentrated processing units, as well as the way perception and knowledge interact. Finally, a brief description of different kinds of learning is presented.

3.1 Visual processing levels

The visual pathway can be decomposed into a series of linked transformations along different channels. Within each channel, the transformations can be structured according to the abstraction of the information they handle and the implementation mechanism they use, as shown in table 3.1.

Motor level
Interpretative level
Analogical level Logical level
Temporal level
Spatial level
Physical level

Table 3.1: A schematic proposal for processing levels in the visual pathway.

In the physical level, light information is converted into an electrical signal to be processed. It is in the periphery of the visual system where the light signals are converted into neural signals, using a logarithmic and band-pass filtering [Lim90] (see figure 3.1). In natural systems, photoreceptors are the transducers which carry out this function. In artificial systems, the acquisition device performs it. Sometimes it is necessary to include in this step some additional techniques to adapt the device to the outside conditions, enhance contrast, equalize, etc.

$$\text{light} \quad \Longrightarrow \quad \log(i) \quad \Longrightarrow \quad H(\Omega_x, \Omega_y) \quad \Longrightarrow \quad \text{neural signal}$$

The earliest transformations on those signals do not involve temporal information, but only spatial relationships, such as local edge detection. These transformations act in a given instant, either on a single image or on a couple of them when binocularity is considered. Thus, according to the number of images to be considered, we can distinguish between simple spatial transformations and complex spatial transformations. In any case, only those values acquired in the same time interval, either in a close neighborhood or a wide area, affect a certain point, but no previous input or output signal is considered in the transformations.

The comparison of several frames in the input channel makes it possible to extract some temporal evolution of the impinging signal, such as motion detection. The primitives extracted in this level are crucial for studying the behavior of the elements in the environment. In this case, only a certain time interval is considered and the closer the time instant, the more significant its influence is. In some cases, what is relevant is the change in time, while in other cases, a temporal integration is performed to smooth the evolution.

Higher processing levels include those tasks which are mostly related to intelligence and in which a certain learning and previous experience may be required. Firstly, the logical level makes use of a set of rules which produce new conclusions when their conditions are fulfilled. These rules may consist of different kinds of logical elements such as fuzzy logic, or simple deterministic conduct guides.

Afterwards, the analogue level compares the scenario with those previously studied in order to extract consequences from past experiences. Similarity plays an important role in case-based reasoning, as discussed in [HR01], where it is tackled from many different points of view. Sometimes, the interaction between these two levels helps the creation of new rules from a series of cases, or the extraction of metacases from rules in the base of cases. Thus, the ordering and limit between logical and analogue levels are sometimes fuzzy.

The interpretative level relates the different kinds of information extracted by the different senses to give a meaning to what the individual, natural or artificial, is perceiving. Initially, the interpretation is carried out by each sense independently, and then an intersensory analysis is executed.

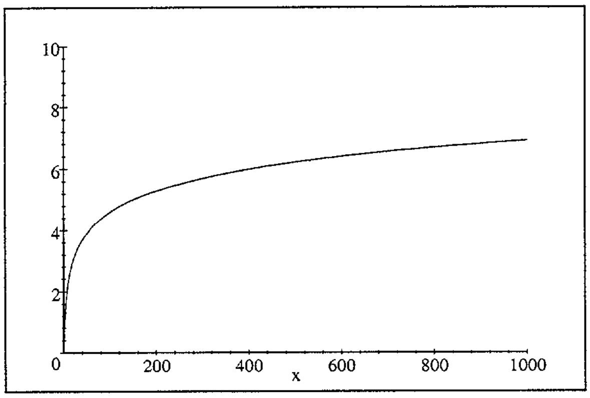


Figure 3.1: Light intensity filtering.

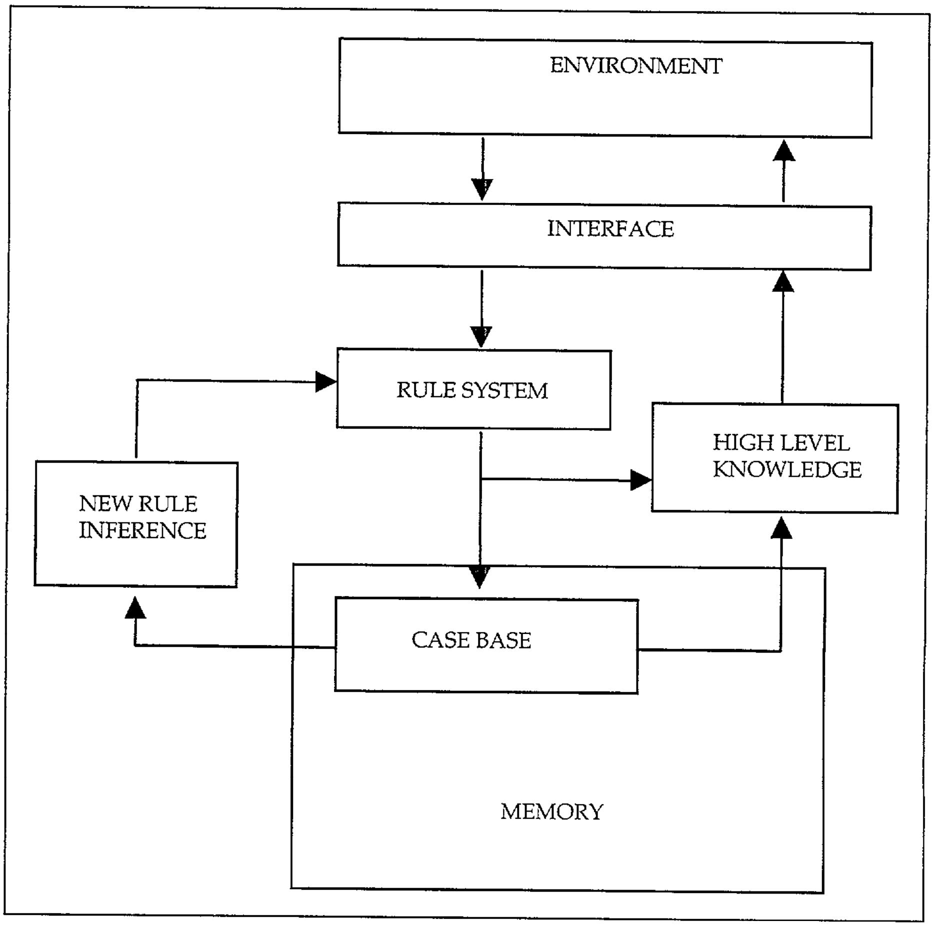


Figure 3.2: Visual processing system scheme.

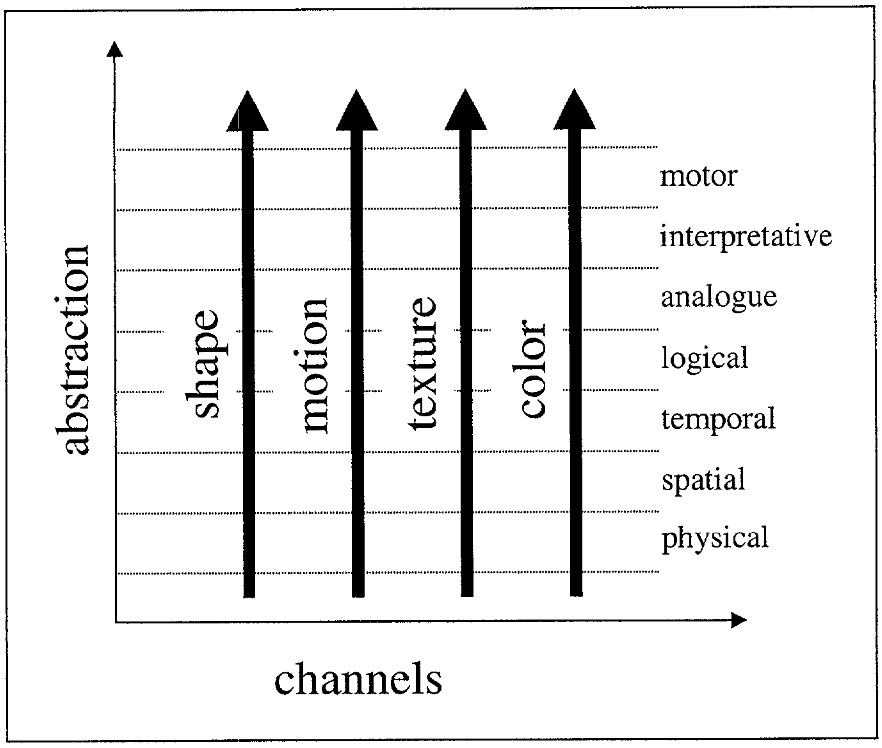


Figure 3.3: Two-dimensional representation of transformation space.

Finally, the motor level makes decisions to give orders to the motor neurons and react to the events of the environment, i.e. it provides the output of the system as a whole.

Thus, a two-dimensional space can be used for representing the transformations in the visual processing system, as shown in figure 3.3. The horizontal axis represents the semantic meaning given to the information provided by each channel, and the vertical axis represents the increasing degree of abstraction achieved by each transformation. Figure 3.2 shows a general scheme of the interaction between different subsystems.

3.2 Relative complexity of the primitives

Two extreme positions are compared below to consider the consequences of the different degrees of complexity that the functions implemented in a certain level may have. We must decide between building *meganeurons* which are able to extract high abstraction features, or combining multiple *microneurons* in a defined structure.

The processing of information through multiple low-abstraction channels results in the following advantages when compared to the use of sparse high-abstraction channels:

- Higher modularity, due to the combination possibilities of the initial operations.
- Reuse of the information which can be important for different tasks.
- Robustness of the whole system since it is easier to build redundant elements to assure the functionality of the most important factors.
- Gradual complexity of transformations, which increases the abstraction in different levels.
- Validation capabilities in intermediate steps.

On the other hand, performing complex operations separately from the beginning results in the following advantages:

- Lower number of communication pathways due to the more reduced interaction.
- Lower coordination requirements.
- Possibility of assigning priorities to some channels due to their independence.
- Faster abstraction of information.
- Disabling and enabling capabilities for certain channels when alert situations require sacrificing some information in favor of some other.

This comparison in the perceptual systems is analogue to the discussion CISC (complex instruction set computer) - RISC (reduced instruction set computer) in the field of computer architecture. The advantages of these two schemes are the following [Zar96]:

RISC:

- Higher speed.
- Lower cost.

CISC:

- Direct support to high level languages.
- Migration of software functions to hardware.
- Compatibility upwards, which makes the number of instructions increase.

Complementary to this discussion about the complexity of the primitives, the work by Leibovic [Lei88] presents the relevance of convergence and divergence in information processing, since the way the functions are implemented results in a certain scheme for the converging and diverging nodes where the information is processed.

3.3 Perception and knowledge

Our universe is a whole, but at the same time, it is an aggregation of many entities that interact with each other and that can be studied as wholes themselves. Natural systems can be perceived as hierarchies that can be decomposed in subelements that perform a certain function into the system with autonomy to carry it out, but that can also be seen as a whole, and are thus decomposable into subparts which aggregate to build a higher level. Individuals consist of organs, which are made up of tissues, which are formed by cells, etc. In the same way, countries are composed of regions, with neighborhoods of families with individuals, etc. In this hierarchy, it is necessary to follow an abstraction process that allows considering groups of elements that interact as a whole, when we go from bottom to top, as well as an analysis process to decompose elements into smaller elements, when we go from top to bottom. This synthesis-analysis process makes it easier to understand and represent the world and, at the same time, provides an effective way of distributing the tasks when a certain function is to be carried out. Each element in this structure is a whole and a part, that is to say, what Koestler calls a *holon* (from Greek *holos*: whole) [Koe78].

Also in industrial and technological fields, there are some structures that are commonly used and that follow the holon idea. Examples of that are cellular manufacturing, object oriented software engineering or holonic robotic systems. In cellular manufacturing, elements are associated according to their outputs, not to the similarity of the functions they carry out. They are autonomous and flexible in performing these functions and adapting to new ones as far as they fulfill the task they are in charge of. In object oriented software engineering, data and procedures are encapsulated, and polymorphism allows adaptability to different situations and generalization in the building of functions. Elements are structured in classes and subclasses, being inheritance present in the way from upper levels to lower ones. In holonic robotic systems, information is processed locally and shared once it has been transformed. Not only the functions are decentralized but also the information processing and there is a recursion in the assignment of functions.

3.3.1 Abstraction in the visual pathway

Our knowledge does not work in terms of pixels, photons, electrical impulses, etc., but we need to extract, from the environment, those features that are useful for us.

The utility of this information is given by our needs, whatever the kind: food, danger, association, etc. That implies a process of abstraction from lower level information to understandable representations of the world outside. Our visual perception is based on the identification of the elements and their description as objects with properties and procedures, as in an object oriented programming language. That, sometimes, implies a loss of information, in the sense that the transformation cannot be performed in the opposite direction in a complete way, that is to say, recovering all the information that was present at the input. But this organization of the information into a more structured description means that single inputs do not represent anything by themselves, but into the whole, and it is not necessary to be aware of every single signal that our senses perceive, but of the meaning of the whole input flow.

As we abstract, single signals are somehow scanned and transformed into more significant signals, the *scanners*, while in the process of decomposition, highly semantic signals, the *triggers*, make other signals appear as if they were condensed into it [Koe78]. The visual processing that takes place from our retina to our brain builds a hierarchy from the single inputs we receive. It is a bottom to top process. At the beginning, we have only photoreceptor outputs, as pixels in a picture. The mixture of the outputs they generate builds colors. The combination of different color units makes lines, borders, uniform surfaces, and so on. These elements constitute objects and these objects are parts of a scene.

As well as we can examine the composition of the scene, we could examine how these elements are extracted, and build a holarchical structure for the units that must analyze the picture or the sequence of pictures. Each feature to be analyzed corresponds to a holon in a certain level and not only the aggregation of these units but also the combination and contrast of them make it possible to extract higher level patterns.

When trying to analyze an image, we intend to give names to the things we see. Is the language guided by the knowledge or is the knowledge constrained by the language? We build our language as a means of expression and, particularly, of description. However, in order to be practical, we associate elements in classes and try to abstract groups of elements, even if these groups are established according to our need to be specific, but, at the same time, general. There are two processes, one of segmentation, and another one of relation and integration.

The identification of the elements that are present in a scene can be carried out in two ways:

- A case-based identification
- A rule-guided identification

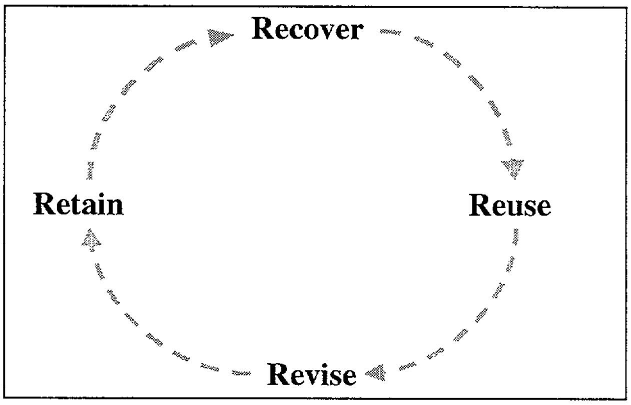


Figure 3.4: Case cycle for a case-based reasoning system.

The first one consists in searching for the closest element to the one we analyze among those which have been recognized before. This is what happens when we try to extract the color of an object. Possibly, we have never perceived this wavelength before, but we describe it with the name that we use for the closest wavelength we know. It is then necessary to define a certain measure of distance that indicates us how close two objects are. And even a certain threshold that determine when two values are far enough to be considered as different and thus belonging to different classes. Cases in a case-based reasoning follow the cycle shown in figure 3.4.

The second one is based on following a procedure that determines a certain feature of the object, such as, for example, the number of angles of a square or a triangle. Here, what we need is a series of instructions that guides us in the identification, generating an output that classifies the object. It is also necessary to define when an output is not classifiable into one of the groups already learned and may represent a new record in our scheme. These topics are more deeply discussed in [Rod96].

When a new entity cannot be classified because it is far from other entities or it does not fulfil certain criteria, our holarchy becomes wider. Sometimes it creates a new branch, and sometimes it unifies several ones into a higher level group.

According to the situation we must deal with, we use a different level in our hierarchy: sometimes we work with high level concepts, with a high level of abstraction, and sometimes we must go down to the underlying levels which may explain the basis of the upper concepts.

At the same time that we abstract structures from the outside world, there is a transformation in the entropy of the information processed. As we go up in our representation system, that is to say, as we abstract, detailed information is no longer used, but those elements that have been identified and classified. Thus, there is an increase in the semantic value of this information as well as a decrease in the completeness of the information, in the sense that it is not possible to go back in the transformation.

3.3.2 Hierarchical systems: holarchies

Our representation of the world is a holarchy, i.e. a hierarchy of holons, where each holon is an object with properties and procedures. Properties are obtained from case-based identification, while procedures determine the possible transformations that can be carried out on these objects. Thus, holarchies are good mechanisms for our knowledge to work, since abstraction, up to a certain point, makes it easier to understand our environment.

If we wish to represent a holarchy, it is necessary to build a tool to define holons. We must design them in terms of properties, procedures and information flows, or interaction that may be produced between holons:

Firstly, it is necessary to distribute the tasks that are going to be performed within each holon. This distribution must not be made according to the way parts work or to the similarities of the functions they performed, but to the output they produce. Holons integrate themselves to achieve a common objective. How this objective is fulfilled is a question of the following level.

Secondly, we must establish the control functions, such that there is a supervision of what is being done, from a higher level. Even if holons are autonomous to determine how a task is performed, there must be some checking points on the outputs they generate.

Finally, it is also necessary to determine the flows that allow holons to interact. Taking into account that holons integrate to build new larger holons, a connection among them must exist. Besides these interholon connections, there must also be intraholon connections, and a global hierarchical communication.

These three steps are repeated in an iterative loop until a satisfactory scheme is reached. Holons are aggregated to achieve a common objective and the cooperation of different subunits is what provides successful results. The whole is not only the addition of its parts: four equal segments do not make a square unless they are disposed in a certain way. Each subholon, being a holon, has flexibility to adapt to new tasks when it is required by the global system.

In this sense, the behavior of a holon is defined by a *canon* and a *strategy.* The canon determines the fixed properties of the holon, that is to say, the framework into which the holon must work. On the other hand, the strategy defines how the holon will react to the different situations that may be presented to it. The canon sets the general rules while the strategy determines the behavior in real time.

3.4 Learning

There are three faces in a holarchical structure: a local environment, an intraholon relationship and a global structure. As well as the flows of information can be produced within a holon, between neighbor holons or into the whole structure, the process of learning can also happen at three levels:

- First order learning is the one that happens within a holon. By means of completing the operation functions with memory functions it is possible to adapt the way holons reach their objectives and improve the way the tasks are executed.
- Second order learning is the one which appears from the cooperation of holons. This allows creating new and more complex functions from the combination of simpler ones.
- Third order learning is based on a change in the structure and is performed by certain non-operational holons whose function is the supervision of the operational ones. That is to say, they perform a change in the way the other holons work, thus adapting the structure to new situations.

The lower levels in transformation hierarchy have a very reduced range of possibilities for learning since their relation to physical implementation makes them more rigid. It is in the higher levels where experience allows increasing, improving and speeding the functionality of the system.

3.5 The role of time in the visual processing

We could consider time as one more coordinate in our visual processing system. Thus, we could work with two or three spatial dimensions and one temporal dimension. However, it has some peculiarities which are worth considering. With spatial dimensions, we can go forward or backward and we can compare regions in any direction. In time, only one direction is possible. At most, we can store in our finite memory what has happened during the last period, and only inside this buffer can we access randomly. Inside any world with which we can communicate, the direction of time is uniform [Wie60].

If time is not one-directional and has not a scale equivalent to ours, we cannot establish cause-effect relations, since these components would be inverted, even if we could justify the causes by means of the effects, and prediction would be transformed into a diagnosis. Time, according to Newton, is reversible. We can make a system evolve according to its laws forward as well as backward. However, for Bergson, time is one-directional and there is no turning back in its evolution. The same laws cannot be applied when we progress forward and backward.

Taking into account that not all transformations are bijective, the inputs will not always be available in the information provided by the outputs. At least, they should be injective. However, it is possible to contract time in such a way that different instants can be considered as simultaneous, in a certain kind of temporal integration.

Chapter 4

NEURON-LIKE DISCRETE FILTERS: NEWTON FILTERS REVISITED

In this chapter, we describe the initial elements we will use for the estimation of orientation. This is the basis for the object recognition, motion analysis and texture classification techniques which will be explained in the following chapters. They are inspired in Newton filters but introduce some modifications which make them more suitable for the purposes which have been described.

Many sets of filters have been proposed to locate the edges of an object, like those presented by Prewitt, Sobel, Robinson or Kirsch [SHB99], or those used by Moreno [Mor93], which have been the starting point for the modified Newton filters. However, some further information is often required and the detection itself does not allow a higher level analysis of the scene. Thus, the characterization of edges according to their orientation provides much more information about the shapes which are studied.

This work presents a new set of tools, the modified Newton filters, and their applications to edge orientation estimation, shape representation and motion analysis. From Newton filters, which have been previously used as neuron-like structures for retinal processing by Quesada et al. [QAM99], a new set with some advantageous features is built. We show how these filters are obtained and how they are used for estimating edge orientation.

4.1 Newton filters

Newton filters are tools based on the repeated use of simple binary operations to build more complex ones, which can compute a wider range of inputs. If we use addition and subtraction of two real or integer numbers as basic operations, we can combine them in different layers in such a way that the resulting functions are linear combinations of the inputs. Each layer operates on the ordered set of results of the previous layer and the operation which is carried out within a layer is the same for each unit inside it, regardless of the position where it is performed. Nevertheless, the operations of different layers may vary. We explain below the most relevant mathematical properties of this kind of filters, since they are important for the next chapters. Firstly, we can relate Newton filters to difference equations, as ahown in figure 4.1.

$$\frac{dY(x)}{dx} = F(x)$$

$$Y(x+1) - Y(x) = F(x)$$

$$Y(x) = Y(x-1) + F(x-1)$$

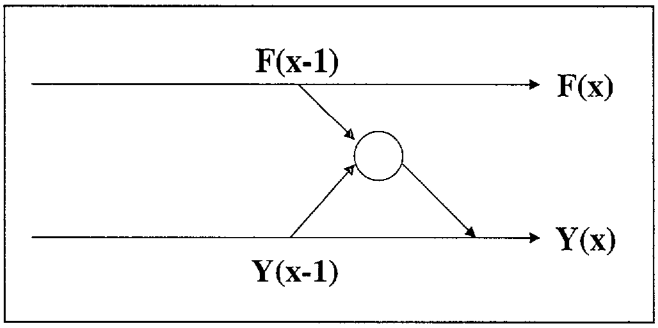


Figure 4.1: Difference equation.

One of the features of Newton filters is the fact that a change in the order of the operations, i.e. of the layers, does not affect the final result, in such a way that a filter is completely identified by the number of additive and subtractive layers it contains, regardless of the order in which they are arranged. Thus, $NF(A_m, D_n)$ is the filter which computes $m+n+1$ inputs, with m additive and n difference layers, as shown in the example in figures 4.2 and 4.3. It is also important to emphasize that, if a filter contains at least one subtractive layer, the sum of all weights is 0 and, therefore, when it operates on a constant input signal, the output will be 0.

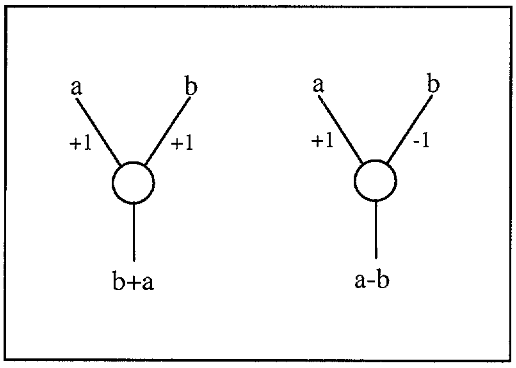


Figure 4.2: Basic operations of Newton filters.

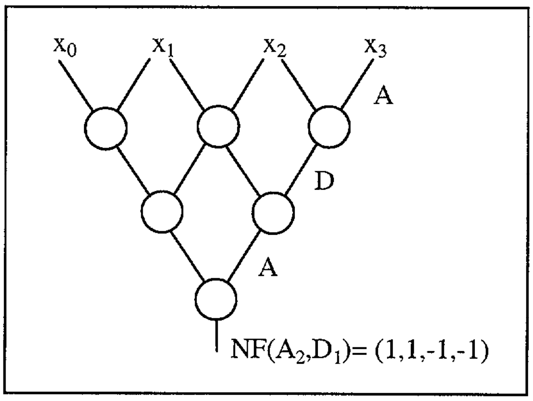


Figure 4.3: Example of a Newton filter with two additive layers and one subtractive layer.

Moreover, the number of subtractive layers determines the number of zero-crossings in the resulting weight vector. Since we can build L different filters for L inputs, and they constitute a linearly independent set of vectors, we can code a discrete signal of L elements by applying these L filters to the signal. If we build the corresponding LxL Newton matrix by setting the weights of each filter within a different row, we obtain the following kind of transformations:

$$Ax = \begin{pmatrix} a_{0,0} & a_{0,1} & \cdots & a_{0,L-1} \\ a_{1,0} & a_{1,1} & \cdots & a_{1,L-1} \\ \cdots & \cdots & \cdots & \cdots \\ a_{L-1,0} & a_{L-1,1} & \cdots & a_{L-1,L-1} \end{pmatrix} \begin{pmatrix} x_0 \\ x_1 \\ \cdots \\ x_{L-1} \end{pmatrix} = \begin{pmatrix} f_0 \\ f_1 \\ \cdots \\ f_{L-1} \end{pmatrix} = F$$

Where $(a_{k,0}, a_{k,1}, ..., a_{k,L-1})$ are the weights of the k^{th} filter, $(x_0, x_1, ..., x_{L-1})$ are the inputs and $(f_0, f_1, ..., f_{L-1})$ are the results of the L filters applied to these values. On the assumption that the output is constant after having computed N layers, i.e. the input signal corresponds to a polynomial of the N^{th} degree, these filters could be used as a prediction tool, using an autoregressive scheme as those described by [Ale86] (see figure 4.5):

$$x(n) = \sum_{k=1}^{N} a_k x(n-k) + v(n)$$

In addition, the square of one of these Newton matrices is a diagonal matrix, and all values into the diagonal are 2^{L-1}, where L is the length of the filters. Thus, its inverse is the matrix itself, but multiplied by 2^{1-L}, and that makes the process to recover the original signal very similar to that followed to apply Newton filters:

$$\begin{array}{lcl} AA = 2^{L-1}I & \Longrightarrow & A^{-1} = 2^{1-L}A \\ AX = F & \Longrightarrow & A^{-1}AX = A^{-1}F \end{array}$$

$$X = 2^{1-L}AF$$

For example, for five-element inputs, filters must consist of 4 layers and we could build 5 filters with 4, 3, 2, 1 and 0 additive layers, whose weights are shown below:

$$NF(A_4, D_0) = (1, \ 4, \ 6, \ 4, \ 1)$$

$$NF(A_3, D_1) = (1, \ 2, \ 0, -2, -1)$$

$$NF(A_2, D_2) = (1, \ 0, -2, \ 0, \ 1)$$

$$NF(A_1, D_3) = (1, -2, \ 0, \ 2, -1)$$

$$NF(A_0, D_4) = (1, -4, \ 6, -4, \ 1)$$

As a result, the following transformation matrix is obtained:

$$A_5 = \begin{pmatrix} 1 & 4 & 6 & 4 & 1 \\ 1 & 2 & 0 & -2 & -1 \\ 1 & 0 & -2 & 0 & 1 \\ 1 & -2 & 0 & 2 & -1 \\ 1 & -4 & 6 & -4 & 1 \end{pmatrix}$$

And its inverse is:

$$A_5^{-1} = 2^{1-5} A_5 = \begin{pmatrix} \frac{1}{16} & \frac{1}{4} & \frac{3}{8} & \frac{1}{4} & \frac{1}{16} \\ \frac{1}{16} & \frac{1}{8} & 0 & -\frac{1}{8} & -\frac{1}{16} \\ \frac{1}{16} & 0 & -\frac{1}{8} & 0 & \frac{1}{16} \\ \frac{1}{16} & -\frac{1}{8} & 0 & \frac{1}{8} & -\frac{1}{16} \\ \frac{1}{16} & -\frac{1}{4} & \frac{3}{8} & -\frac{1}{4} & \frac{1}{16} \end{pmatrix}$$

We can combine one-dimensional filters to build a new set of structures which operate on two-dimensional data. In a work by Quesada et al. [QAM99], two-dimensional filters are created considering additions and subtractions of 4 elements as basic operations, with the constraint that the global weight must be 0 if there are subtractions. The seven filters which are obtained, shown in table 4.1, do not constitute a complete set, as it is not possible to recover the original information from the results.

This mechanism to build two-dimensional filters reduces the possible combinations and generates incomplete sets of transformations. The multiplication of horizontal and vertical filters, which is equivalent to applying a two-dimensional filter repeatedly, does generate a complete set. For instance, with a 3x3 pattern, we first calculate all three-component one-dimensional filters. We can include two, one or no additive layer, and the others will be subtractive. That yields the following filters:

$$NF(A_2, D_0) = (1, \ 2, \ 1)$$

$$NF(A_1, D_1) = (1, \ 0, -1)$$

$$NF(A_0, D_2) = (1, -2, \ 1)$$

Next, we combine these 3 filters in the horizontal and vertical directions and the set resulting is shown in table 4.2. According to the function they perform, the first one is a low-pass filter, and it is the only filter in which the sum of all weights is not 0, as it contains no subtractive layer in either direction. However, all others have compensated weights as they have subtractive layers, and the function associated to each one of them is an approximation of a derivative of the original signal. The order of the derivative is given by the number of subtractive layers it contains, and the orientation is determined by the combination of horizontal and vertical differences which are carried out. Therefore, we can relate filters E_i to approximations of the following derivatives of the original input data:

$$\begin{array}{cc}
E_1 \Longrightarrow & -\frac{\partial u}{\partial x} \\
E_2 \Longrightarrow & \frac{\partial u}{\partial y} \\
E_3 \Longrightarrow & -\frac{\partial^2 u}{\partial x \partial y} \\
E_4 \Longrightarrow & \frac{\partial^2 u}{\partial y^2} \\
E_5 \Longrightarrow & \frac{\partial^2 u}{\partial x^2} \\
E_6 \Longrightarrow & -\frac{\partial^3 u}{\partial x \partial y^2} \\
E_7 \Longrightarrow & \frac{\partial^3 u}{\partial x^2 \partial y} \\
E_8 \Longrightarrow & \frac{\partial^4 u}{\partial x^2 \partial y^2}
\end{array}$$

Furthermore, this set of filters constitutes a complete transformation from the point of view of inversion, i.e. information is preserved, and the original values could be recovered from the outputs they provide. If we represent the weights of these filters as a matrix in which each row corresponds to a filter and each column is associated to a position in the filter, i.e. a position in the neighborhood of the point on which they are applied, we obtain the following matrix.

$$A = \begin{pmatrix}
1 & 2 & 1 & 2 & 4 & 2 & 1 & 2 & 1 \\
1 & 0 & -1 & 2 & 0 & -2 & 1 & 0 & -1 \\
1 & 2 & 1 & 0 & 0 & 0 & -1 & -2 & -1 \\
1 & 0 & -1 & 0 & 0 & 0 & -1 & 0 & 1 \\
1 & 2 & 1 & -2 & -4 & -2 & 1 & 2 & 1 \\
1 & -2 & 1 & 2 & -4 & 2 & 1 & -2 & 1 \\
1 & 0 & -1 & -2 & 0 & 2 & 1 & 0 & -1 \\
1 & -2 & 1 & 0 & 0 & 0 & -1 & 2 & -1 \\
1 & -2 & 1 & -2 & 4 & -2 & 1 & -2 & 1
\end{pmatrix}$$

The linear independence of Newton Filters makes it possible to use them as a codification of a signal. Figure 4.4 shows an example of an image which has been coded by applying Newton filters by columns, thus replacing every 32 values in a column by the resulting values of applying the 32 linearly independent Newton filters of length 32 on them.

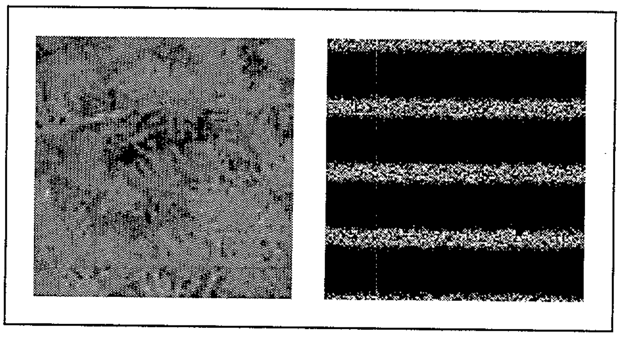


Figure 4.4: Image codification by columns using Newton filters.

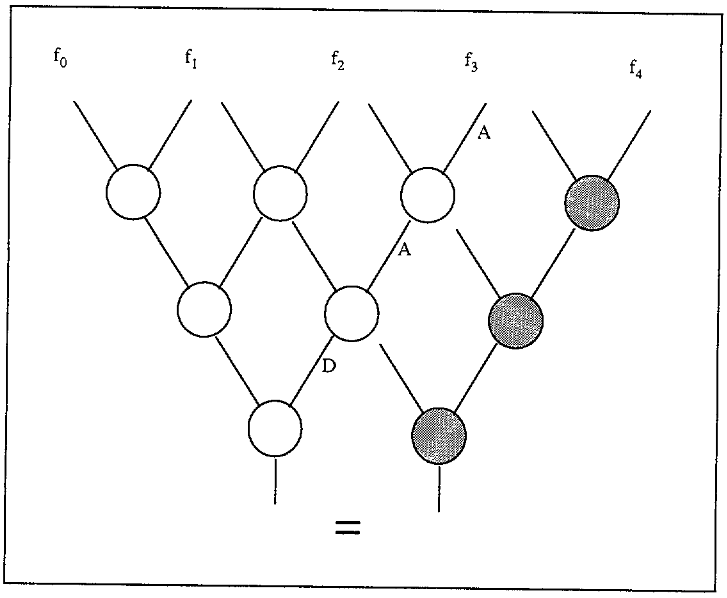


Figure 4.5: Autoregressive prediction using Newton filters.

Experiments on the activity of ganglion cells of higher vertebrate retina show that these cells perform computations on their receptive fields, i.e. on the local neighborhood where the presence of light stimuli affects their activity, which are similar to those described by these filters. Therefore, the global operation on the images falling upon the retina can be considered as a convolution with this kind of operators.

The basic operations to be performed within each layer of a Newton filter can be generalized by the use of linear combinations of every couple of values affected. This leads us to the so called generalized Newton filters, whose layers compute the operations shown in equation (4.1). Let x_n^i be the i^{th} value of the n^{th} layer, and let a_n and b_n be the coefficients used in the n^{th} layer, the values of each layer are calculated as follows:

$$x_n^i = a_n x_{n-1}^i + b_n x_{n-1}^{i+1} = a_n \left(x_{n-1}^i + e_n x_{n-1}^{i+1} \right) \tag{4.1}$$

When all a_i are 1, the filter is said to be normalized. The global output y of one of these filters is given by the expression:

$$y = x_0 + x_1(e_1 + e_2 + ... + e_L) + x_2(e_1 e_2 + e_1 e_3 + ... + e_{L-1} e_L) + ...$$
$$... + x_{L-1}(e_1 e_2 e_3 ... e_L)$$

$$\boxed{\begin{matrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{matrix}} \qquad N_0 : \begin{pmatrix} + & + \\ + & + \end{pmatrix}$$

$$\boxed{\begin{matrix} 1 & 0 & -1 \\ 2 & 0 & -2 \\ 1 & 0 & -1 \end{matrix}} \qquad N_1 : \begin{pmatrix} + & - \\ + & - \end{pmatrix}$$

$$\boxed{\begin{matrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{matrix}} \qquad N_2 : \begin{pmatrix} - & + \\ - & + \end{pmatrix}$$

$$\boxed{\begin{matrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{matrix}} \qquad N_3 : \begin{pmatrix} + & + \\ - & - \end{pmatrix}$$

$$\boxed{\begin{matrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{matrix}} \qquad N_4 : \begin{pmatrix} - & - \\ + & + \end{pmatrix}$$

$$\boxed{\begin{matrix} 1 & 0 & -1 \\ 0 & 0 & 0 \\ -1 & 0 & 1 \end{matrix}} \qquad N_5 : \begin{pmatrix} + & - \\ - & + \end{pmatrix}$$

$$\boxed{\begin{matrix} -1 & 0 & 1 \\ 0 & 0 & 0 \\ 1 & 0 & -1 \end{matrix}} \qquad N_6 : \begin{pmatrix} - & + \\ + & - \end{pmatrix}$$

Table 4.1: Original two-dimensional Newton filters.

$$\boxed{\begin{array}{rrr}1&2&1\\2&4&2\\1&2&1\end{array}}$$

E_0 :
(1,2,1)^t(1,2,1)

$\boxed{\begin{array}{rrr}1&0&-1\\2&0&-2\\1&0&-1\end{array}}$ E_1 : (1,2,1)^t(1,0,-1)	$\boxed{\begin{array}{rrr}1&-2&1\\2&-4&2\\1&-2&1\end{array}}$ E_5 : (1,2,1)^t(1,-2,1)
$\boxed{\begin{array}{rrr}1&2&1\\0&0&0\\-1&-2&-1\end{array}}$ E_2 : (1,0,-1)^t(1,2,1)	$\boxed{\begin{array}{rrr}1&0&-1\\-2&0&2\\1&0&-1\end{array}}$ E_6 : (1,-2,1)^t(1,0,-1)
$\boxed{\begin{array}{rrr}1&0&-1\\0&0&0\\-1&0&1\end{array}}$ E_3 : (1,0,-1)^t(1,0,-1)	$\boxed{\begin{array}{rrr}1&-2&1\\0&0&0\\-1&2&-1\end{array}}$ E_7 : (1,0,-1)^t(1,-2,1)
$\boxed{\begin{array}{rrr}1&2&1\\-2&-4&-2\\1&2&1\end{array}}$ E_4 : (1,-2,1)^t(1,2,1)	$\boxed{\begin{array}{rrr}1&-2&1\\-2&4&-2\\1&-2&1\end{array}}$ E_8 : (1,-2,1)^t(1,-2,1)

Table 4.2: Expanded two-dimensional Newton filters.

4.2 Divisional inhibition and illumination changes

A phenomenon which has been observed in the functioning of motor and sensory neurons, and which has a great importance when studying the combined operation of a group of them, is the presynaptic inhibition. It consists in a decrease in the output of a neuron when one of its neighbors undergoes a great excitation. The fact that the inhibition is produced before the synapse causes the output not to be transmitted to the following layer with its original value, but after being inhibited. The ideal expression is shown in equation (4.2), where N is the activity of the inhibiting neighbor and x_0 and x_1 determine the range of values for which inhibition is considered as linear.

$$I = \begin{cases} 1 & if \;\; N \leq x_0 \\ aN + b & if \;\; x_0 < N < x_1 \\ \infty & if \;\; N \geq x_1 \end{cases} \tag{4.2}$$

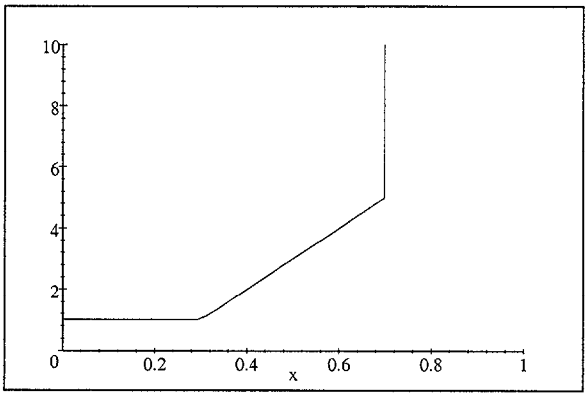


Figure 4.6: Ideal divisional inhibition.

Different types of models have arisen. Among them is the model proposed by Lettvin [Let62], who called it *linear divisional inhibition*, and in which inhibition consists in a change in the electrical conductivity according to equation (4.3).

$$A = \frac{E}{1 + \frac{I}{I_0}} \tag{4.3}$$

The resulting activity A is obtained from the reaction E of the neuron considered and the inhibition I it receives, being I_0 a constant which regulates the inhibitory

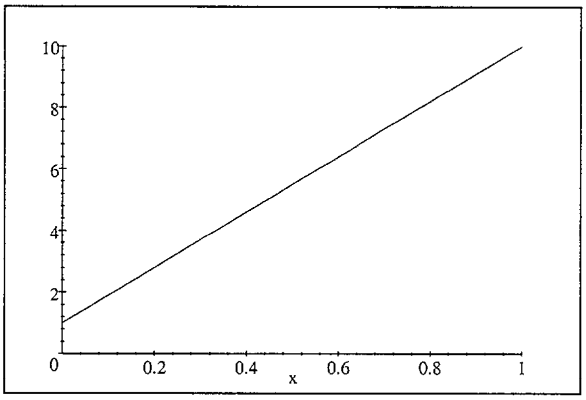


Figure 4.7: Linear divisional inhibition.

action. Other models of divisional inhibition have been proposed since then, such as the exponential model [MQA00], which resembles more the ideal function and does not cause so many problems with extreme values as it eliminates the linearity of Lettvin's model. As can be observed in equation (4.4), two constants, a and b, are used to regulate the effects of inhibition.

$$A = \frac{E}{\frac{I}{a}e^{\frac{I}{b}} + 1} \tag{4.4}$$

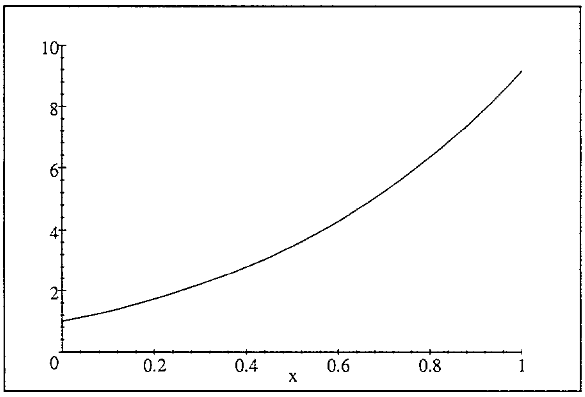


Figure 4.8: Exponential divisional inhibition.

As mentioned above, if Newton filters have at least one subtractive layer, the sum of all weights is zero. This feature is very important when filters are used as edge detectors, as the output will be zero in homogeneous zones, whereas excitations will be produced in those areas with variations. If a global illumination change happens, it will equally affect all outputs and the quotient will remain stable. With linear divisional inhibition and exponential inhibition, this invariance is preserved with constant increases or decreases in the illumination conditions, but not when scaling the values. This is also obtained with a pure divisional inhibition as shown in equation (4.5).

$$A = \frac{E}{I} \tag{4.5}$$

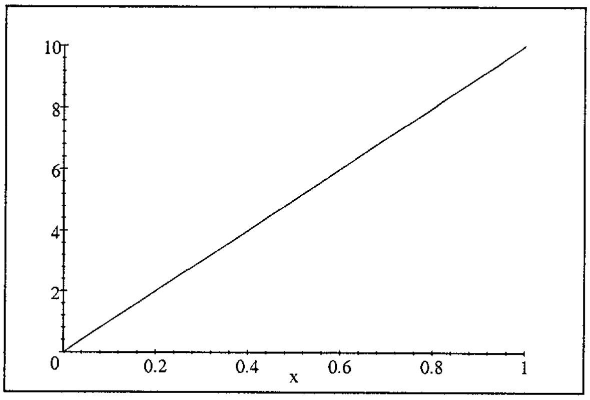


Figure 4.9: Pure linear divisional inhibition.

If a uniform increase occurred in all values of the input signal as shown in equation (4.6), when the filters which are used have null global weights, the increases would be cancelled. Therefore, the proportion of the different reactions remains stable. If we represent the weight of the n^{th} filter in position (i,k) as $w^n_{i,k}$, where $(0,0)$ is the central position, the following applies:

$$\textstyle\sum_{i,k} w^n_{i,k} = 0 \quad i,k \in \{-1,0,1\}$$

$$F_n(x,y) = \sum_{i,k} w^n_{i,k} I(x+i, y+k)$$

$$T_{n,m}(x,y) = \frac{F_n(x,y)}{F_m(x,y)}$$

$$I'(x,y) = aI(x,y) + b \tag{4.6}$$

$$T'_{n,m}(x,y) = \frac{a\sum_{i,k} w^n_{i,k} I(x+i,y+k) + b\sum_{i,k} w^n_{i,k}}{a\sum_{i,k} w^m_{i,k} I(x+i,y+k) + b\sum_{i,k} w^m_{i,k}} = \frac{aF_n(x,y)}{aF_m(x,y)}$$

$$T'_{n,m}(x,y) = T_{n,m}(x,y) \tag{4.7}$$

Where $T_{n,m}$ represents the relation between the n^{th} and the m^{th} filters and $T'_{n,m}$ represents the same proportion as $T_{n,m}$, but applied to the input signal after having changed the global conditions. As can be observed in equation (4.7), the output is the same as it was before the change.

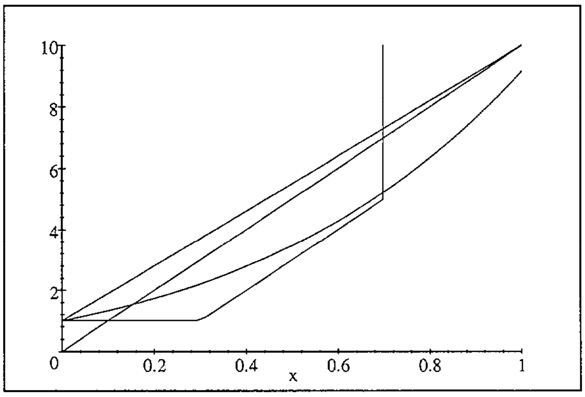


Figure 4.10: Comparison of multiple divisional inhibition approximations.

4.3 Modified Newton filters and edge orientation estimation

In the descriptions of neurophysiologists about the functioning of those cells in charge of extracting information from the visual environment, we find that their reaction is, in many cases, selective to the direction and orientation of the objects. In fact, there are cells which react to the presence of bars located with a certain slope, while the output decreases when the orientation is different, as explained by Hubel [Hub88] and Kandel [Kan81]. On the other hand, Barlow et al. [BHL64] describe cells whose outputs are modulated by motion direction. Hence the importance of, not only locating edges, but also determining their orientation in an accurate way.

The receptive field of a complex cell of the primary visual cortex consists of parallel excitatory and inhibitory regions. The studies on the behavior of these cells show that the position of the stimulus into the receptive field is not so important as its orientation [Kan81]. This could be explained by the association of many shifted simple cortical cells with the same axis orientation [HW62].

If we wished to label the edges according to the outputs provided by the original Newton filters, we could consider whether one of these filters generates a much higher output than the others when applied onto the same area. If this is the case, the edge has an orientation which is very similar to the one corresponding to such filter. A measure of how similar the edge is to one of the orientations described by these filters is given by the following quotient, where f_i represents the output of filter N_i:

$$\frac{\max\{f_i\}}{\sum_{i=1}^{6}|f_i|}$$

When this value is high enough, we are supposed to be on an edge, as one of the filters has a much higher output than the others and its index would help us decide the orientation. Nonetheless, we find filters with different numbers of non-null values and weight magnitudes. As a result, it is difficult to compare their outputs properly, causing the clearness of the outcome to depend on the orientation of the edges. In this paper, we propose a new set of filters which preserve the convenient properties of original Newton filters, but which also avoid some of the undesirable phenomena. Firstly, as we are going to work with 3x3 filters, we calculate the three-component one-dimensional filter that only contains additive layers $NF(A_2, D_0)$:

$$NF(A_2, D_0) = (1, 2, 1)$$

We try to build filters that react to changes in the 8 main orientations. Consequently, it is necessary to use different signs on both sides of the border. It would be desirable that the weights in the central region of the filter were not 0, as happens in the original filters, since it causes the duplication of the contours, even if there is a perfect edge. To keep the global weight equal to 0, we multiply the negative region by 2, thus obtaining the filters shown in figure 4.11 and table 4.3. In particular, filter M_k reacts to changes in the orientation $k\frac{\pi}{4}$.

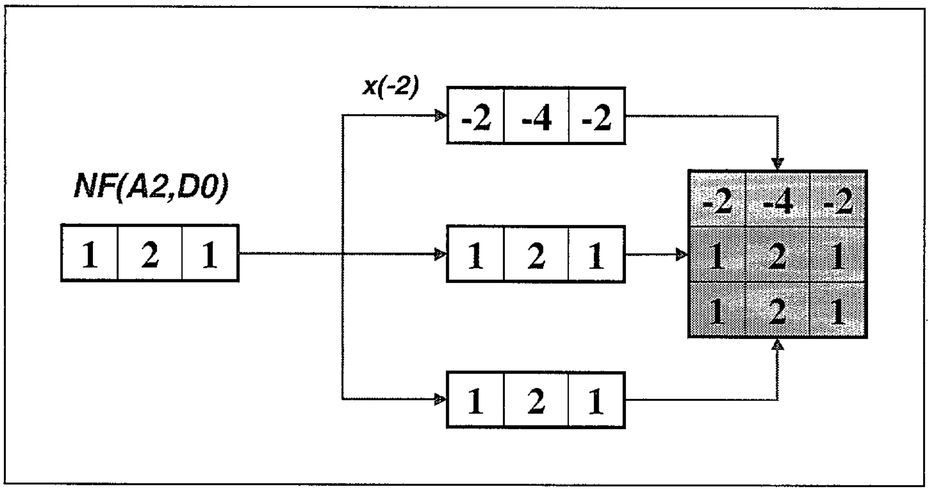


Figure 4.11: Building process of modified Newton filters.

We notice that the output of these filters is independent of the particular grey-value magnitude of the image border, i.e. M_k is invariant under a grey-level translation of the form $I \longrightarrow I + C$, where I is the grey-value of the image and C is any constant. This property is very important because it is well known in perception theory that the relevant information is provided by the relations (differences) between neighbors, rather than the magnitude of the image grey-value. From this point of view, these filters provide a complete representation of the image border up to a grey-level transformation performed by means of a translation in the following sense: We consider a 3x3 neighborhood of an edge pixel and let X be the vector of the grey-value of the 8 neighbor pixels arranged cyclically, let V be the vector $(v, v, v, v, v, v, v, v)^t$ where v is the grey-value of the central edge pixel, and let A be

$\boxed{\begin{matrix} 1 & 1 & -2 \\ 2 & 2 & -4 \\ 1 & 1 & -2 \end{matrix}}$ $M_0 : 0$	$\boxed{\begin{matrix} -2 & 1 & 1 \\ -4 & 2 & 2 \\ -2 & 1 & 1 \end{matrix}}$ $M_4 : \pi$
$\boxed{\begin{matrix} 1 & -2 & -4 \\ 1 & 2 & -2 \\ 2 & 1 & 1 \end{matrix}}$ $M_1 : \pi/4$	$\boxed{\begin{matrix} 1 & 1 & 2 \\ -2 & 2 & 1 \\ -4 & -2 & 1 \end{matrix}}$ $M_5 : 5\pi/4$
$\boxed{\begin{matrix} -2 & -4 & -2 \\ 1 & 2 & 1 \\ 1 & 2 & 1 \end{matrix}}$ $M_2 : \pi/2$	$\boxed{\begin{matrix} 1 & 2 & 1 \\ 1 & 2 & 1 \\ -2 & -4 & -2 \end{matrix}}$ $M_6 : 3\pi/2$
$\boxed{\begin{matrix} -4 & -2 & 1 \\ -2 & 2 & 1 \\ 1 & 1 & 2 \end{matrix}}$ $M_3 : 3\pi/4$	$\boxed{\begin{matrix} 2 & 1 & 1 \\ 1 & 2 & -2 \\ 1 & -2 & -4 \end{matrix}}$ $M_7 : 7\pi/4$

Table 4.3: Modified Newton filters and corresponding orientation.

the 8x8 matrix given by:

$$A = \begin{pmatrix} -4 & -2 & 1 & 1 & 2 & 1 & 1 & -2 \\ -2 & -4 & -2 & 1 & 1 & 2 & 1 & 1 \\ 1 & -2 & -4 & -2 & 1 & 1 & 2 & 1 \\ 1 & 1 & -2 & -4 & -2 & 1 & 1 & 2 \\ 2 & 1 & 1 & -2 & -4 & -2 & 1 & 1 \\ 1 & 2 & 1 & 1 & -2 & -4 & -2 & 1 \\ 1 & 1 & 2 & 1 & 1 & -2 & -4 & -2 \\ -2 & 1 & 1 & 2 & 1 & 1 & -2 & -4 \end{pmatrix}$$

which represents the modified Newton filters coefficients excluding the coefficient associated to the central value (which always equals 2). Then, we can write the relation:

$$AX + 2V = F$$

where $F = (f_0, ..., f_7)^t$ is the modified Newton filters' output vector. The inverse of matrix A is given by (4.8) and the low condition number of these matrices, (see equation (4.9)), means that A provides a nice transformation from a numerical point of view.

$$A^{-1} = \begin{pmatrix} -\frac{11}{48} & -\frac{1}{24} & \frac{1}{16} & -\frac{5}{24} & \frac{5}{48} & -\frac{5}{24} & \frac{1}{16} & -\frac{1}{24} \\ -\frac{1}{24} & -\frac{11}{48} & -\frac{1}{24} & \frac{1}{16} & -\frac{5}{24} & \frac{5}{48} & -\frac{5}{24} & \frac{1}{16} \\ \frac{1}{16} & -\frac{1}{24} & -\frac{11}{48} & -\frac{1}{24} & \frac{1}{16} & -\frac{5}{24} & \frac{5}{48} & -\frac{5}{24} \\ -\frac{5}{24} & \frac{1}{16} & -\frac{1}{24} & -\frac{11}{48} & -\frac{1}{24} & \frac{1}{16} & -\frac{5}{24} & \frac{5}{48} \\ \frac{5}{48} & -\frac{5}{24} & \frac{1}{16} & -\frac{1}{24} & -\frac{11}{48} & -\frac{1}{24} & \frac{1}{16} & -\frac{5}{24} \\ -\frac{5}{24} & \frac{5}{48} & -\frac{5}{24} & \frac{1}{16} & -\frac{1}{24} & -\frac{11}{48} & -\frac{1}{24} & \frac{1}{16} \\ \frac{1}{16} & -\frac{5}{24} & \frac{5}{48} & -\frac{5}{24} & \frac{1}{16} & -\frac{1}{24} & -\frac{11}{48} & -\frac{1}{24} \\ -\frac{1}{24} & \frac{1}{16} & -\frac{5}{24} & \frac{5}{48} & -\frac{5}{24} & \frac{1}{16} & -\frac{1}{24} & -\frac{11}{48} \end{pmatrix} . \qquad (4.8)$$

$$\chi(A) = \chi\left(A^{-1}\right) = \|A\| \left\|A^{-1}\right\| = 5.8284 \qquad (4.9)$$

Therefore, we can recover the vector X as in equation (4.10), which means that we can recover the border grey-values of the image from the modified Newton filters output F, and the grey value v of the central pixel.

$$X = A^{-1}(F - 2V) \qquad (4.10)$$

While in the original Newton filters we observed some differences according to the orientation of the edge we wanted to detect, in this set, the weights are arranged cyclically. This implies that rotating the image a multiple of $\pi/4$ does not alter the magnitude of the output, but only the indices. Figure 4.12 shows the positive outputs of these filters for a circle. A more complete information about the orientation of the edges is given by considering the whole pattern provided by the eight filters. For instance, for 0 and $\pi/2$ oriented edges, we obtain the following outputs:

1	1	0
1	1	0
1	1	0

Input for a 0 oriented edge.

f_0	f_1	f_2	f_3	f_4	f_5	f_6	f_7
8	5	0	−4	−4	−4	0	5

Output for a 0 oriented edge.

0	0	0
1	1	1
1	1	1

Input for a $\pi/2$ oriented edge.

f_0	f_1	f_2	f_3	f_4	f_5	f_6	f_7
0	5	8	5	0	−4	−4	−4

Output for a $\pi/2$ oriented edge.

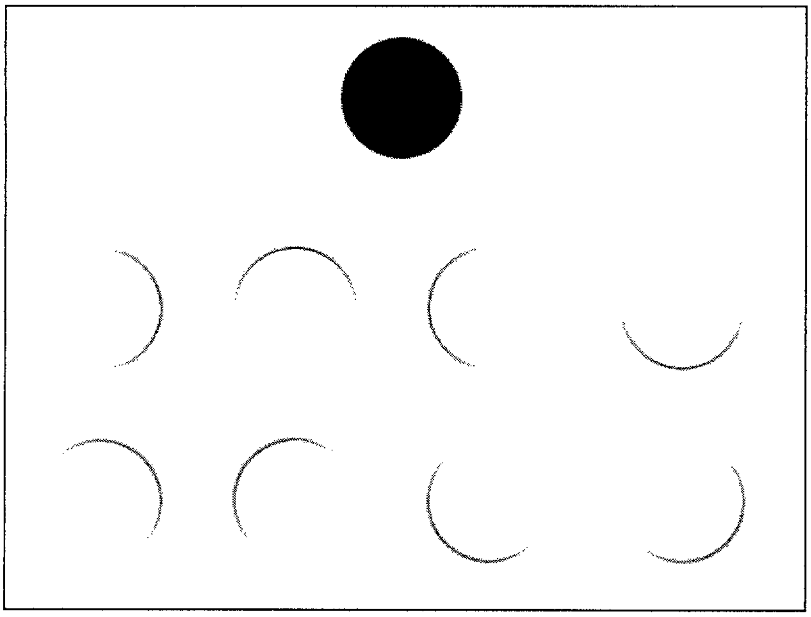


Figure 4.12: Positive outputs of the modified Newton filters for the circle above (the higher the output value, the darker it has been represented).

If the change from one side of the border to the other is higher than one, even if the values may be altered by the correction carried out according to the area covered by the object, the estimation for the orientation remains equal. When the real orientation does not correspond to one of the 8 main directions, as in figure 4.13, we can estimate it by interpolating the higher value in the output vector F with its two neighbors, which provides a much more accurate estimation of the orientation. The interpolating polynomial of the second degree which results is:

$$y = \frac{8(f_{i+1} - f_{i-1}) - 16(f_i - f_{i-1})}{\pi^2}\left(x - \frac{\pi}{4}(i-1)\right)^2 + \frac{8(f_i - f_{i-1}) - 2(f_{i+1} - f_{i-1})}{\pi}\left(x - \frac{\pi}{4}(i-1)\right) + f_{i-1}$$

Where i is the index of the filter with the highest output f_i and positions $i-1$ and $i+1$ are calculated modulo 8. With this polynomial, the maximum value x_{max} is given by:

$$x_{\max} = \frac{4(f_i - f_{i-1}) - (f_{i+1} - f_{i-1})}{2\left[2(f_i - f_{i-1}) - (f_{i+1} - f_{i-1})\right]}\frac{\pi}{4} + \frac{\pi}{4}(i-1)$$

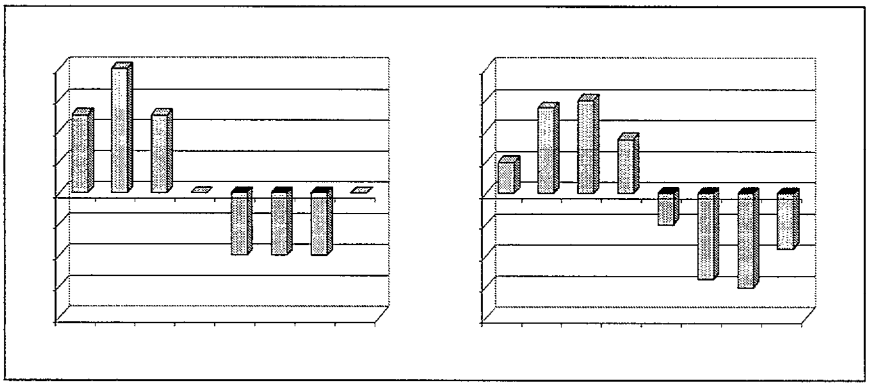


Figure 4.13: Output for a principal orientation (left) and output for a non-principal orientation (right).

For instance, in a non-binary input in which a 1.249 radians orientation is reflected in the values of the pixels according to the proportion of their area covered by the object, as shown in figure 4.14, the output of the filters yields the following result for the estimated orientation. Initially, the maximum value is reached for $i = 2$, i.e. $\pi/2$, and the neighbors to be considered are $i - 1 = 1$ and $i + 1 = 3$:

0	0	0
5/6	1/2	1/6
1	1	1

Numerical values of edge in figure 4.14.

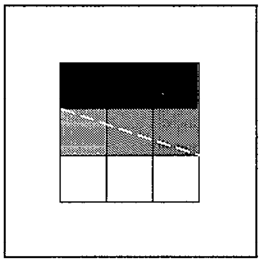


Figure 4.14: Slope 3 edge.

f_0	f_1	f_2	f_3	f_4	f_5	f_6	f_7
2	11/2	6	7/2	−2	−11/2	−6	−7/2

Output of the modified Newton filters for input in figure 4.14.

When interpolation is applied, the following result is obtained:

$$x_{\max} = \frac{4 \cdot 6 - 3 \cdot 11/2 - 7/2}{4 \cdot 6 - 2 \cdot 11/2 - 2 \cdot 7/2} \frac{\pi}{4} + \frac{\pi}{4}(2-1) = \frac{5\pi}{12}$$

By correlating the pattern obtained for a perfect edge with the real one, we can determine how similar the border is to the perfect edges in the 8 main orientations. As the change between both sides of the border may be larger than 1, it is necessary to normalize the output. Let p_i be the i^{th} component of the ideal pattern for a 0 oriented edge, i.e. $P = (8, 5, 0, -4, -4, -4, 0, 5)$, and let F_i be the output of the i^{th} filter, we can calculate the correlation for the 8 main orientations:

$$c_x = \sum_{i=0}^{7} \frac{p_{(i+x) \bmod 8} f_i}{\|P\| \|F\|} \qquad x \in Z_8$$

Nevertheless, as the real orientation may not be one of them, we can interpolate the previous results to test how perfect the edge is for the orientation given by $x_{\max}$, as follows:

$$\begin{aligned} C(x_{\max}) &= \frac{(c_{i+1} - c_{i-1}) - 2(c_i - c_{i-1})}{2} \left(\frac{4x_{\max}}{\pi} - (i-1) \right)^2 \\ &+ \frac{4(c_i - c_{i-1}) - (c_{i+1} - c_{i-1})}{2} \left(\frac{4x_{\max}}{\pi} - (i-1) \right) + c_{i-1} \end{aligned}$$

Similar filters have been proposed by Prewitt, Sobel, Robinson or Kirsch, as described by Sonka et al. [SHB99], but they were not aimed at the simulation of natural systems and cause the duplication of perfect edges (Prewitt and Sobel), are independent of the central value (Kirsch) or may produce the maximum output for imperfect edges (Robinson). Furthermore, in Prewitt and Sobel operators, as well as in the original Newton Filters, oppositely oriented filters have the same output magnitude for any input (only the sign is altered), which means that only a half of them provide some useful information and, therefore, they do not constitute a complete transformation.

The main advantage of this new kind of filters is not the location of edges, but their classification according to their orientation and the invariance under rotations and global illumination changes. These structures may have as a goal, apart from the location of borders itself, the use of this information for other higher semantic level purposes. For instance, when identifying an object, we can use them to build an orientation function of the contour, or when extracting selective motion direction, we can use these filters as basic tools to improve the discrimination.

4.4 A global theoretical framework for information processing in vertebrate retina

In the following chapters of this work, we will present a global scheme which intends to provide a coherent framework for information process in vertebrate retina. There are always gaps in the processing levels from what nature performs and how artificial systems imitate it. However, we try to build a whole scheme, based on simple structures for a multichannel system.

The initial stages are the same for all the channels, since all of them are based on the accurate estimation of edge orientation. Afterwards, the use of this information is different for every channel. In the case of motion analysis, due to the fact that motion is studied from the temporal evolution of the contours, there is a dependence on the shape characterization module.

Even if a separate channel for color can be added, as we will describe in chapter 8, it may also be considered as a part of shape or texture channels, since sometimes they use color information. More detailed information on natural visual systems and examples of their simulation with artificial models are shown in [MQA01].

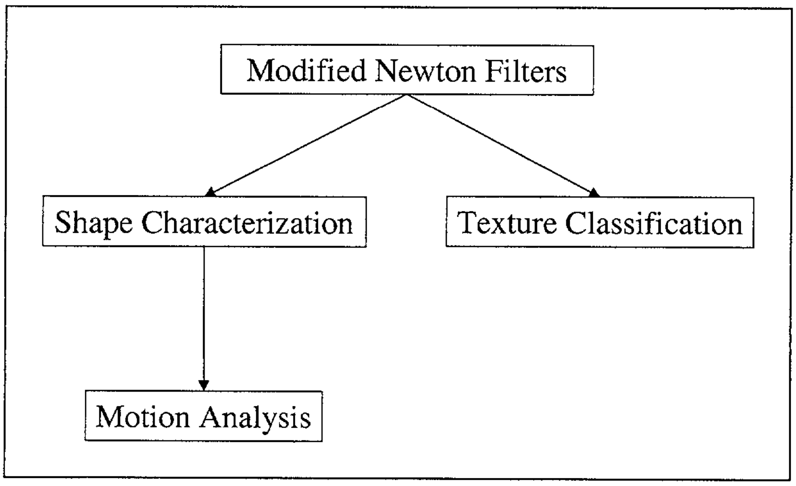


Figure 4.15: Applications of modified Newton filters.

Chapter 5

SHAPE REPRESENTATION

This chapter deals with the problem of identifying an object by means of its contour. The outline is extracted by means of the filters previously described, and an orientation function is built to characterize the shape. We use an energy function to measure the similarity between two objects according to the shapes of their orientation functions.

The outputs which modified Newton filters provide in every point along the contour of an object allow building a one-dimensional representation of its shape, in such a way that we can compare patterns and classify objects according to their outlines. To do this, a Fourier analysis is carried out on the one-dimensional functions which have been generated. In order to generalize our classifying method as much as possible, we have considered the consequences of certain transformations on the objects, as well as the different conditions in which a certain shape can be found. By studying how Fourier coefficients are altered by these transformations, we have determined how the contours could be compared and fitted.

Next, the problem of occlusion is considered in order to characterize partially occluded objects and extract the segments which fit a certain shape. Finally, we present experimental results in which this technique is used to extract groups of similar shapes from a large database containing 1000 images of marine animals.

5.1 Contour-based shape representation

We have described a technique to extract local information about the orientation of the contour of an object. Afterwards, we must combine these results to obtain a global description of a given shape. Below, we describe the representation we will use for the closed contour of an object.

Once we have located the edges and the orientation in every point of the contour, we can build a one-dimensional sequence which describes how the latter changes as we move along the border. If the edges are not clear enough, double borders may appear and must be eliminated before continuing.

We must first select a point of the outline and, according to its orientation, find the neighbor that best corresponds to the edge it indicates. In case none of the neighbors continues the edge in a congruent direction, we must change the orientation as that means we are on a high curvature area. If we continue searching for non-visited neighbors until we close the figure, at the end of the process we must have covered the border of the object and found a representation of its orientation function.

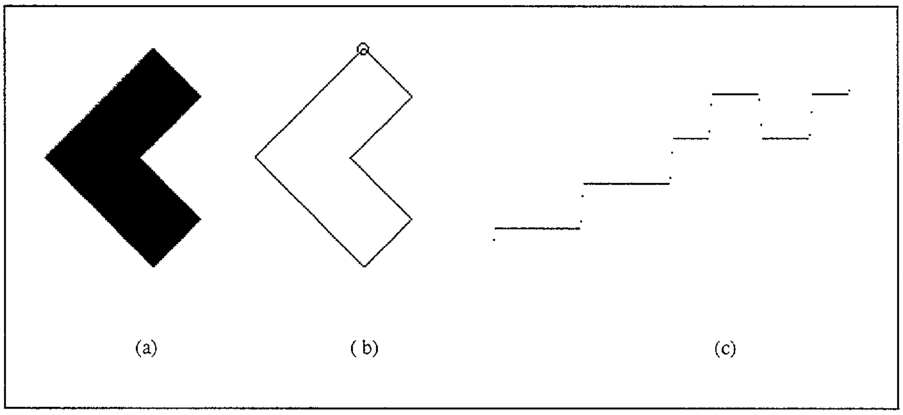


Figure 5.1: (a) Input image. (b) Contour and starting point for orientation function. (c) Orientation function (counterclockwise).

Figure 5.1 shows the outline of an object and how the orientation of the points on the border changes. We could use curvature functions to represent shapes, but straight lines would be represented as 0 while strong changes in the orientation are shown as high positive or negative values, which makes these functions much more sensitive to noise than the functions we use. Different types of one-dimensional shape representations are described by Loncaric [Lon98].

We have seen above how we could extract an orientation function from the outputs of the modified Newton filters applied to an image. Thus, the problem of identifying shapes, i.e. two-dimensional functions, has been reduced to the association of two one-dimensional functions, one considered as the reference pattern, and the other one obtained from the situation which has been presented. In order to provide these functions with a certain continuity, we forced some conditions, such as selecting values of the orientation which do not differ in more than π radians from its neighbors, except for the first and the last points of a closed curve, which are actually neighbors. No condition has been set on the starting point, since having the whole object in the image generates a closed curve. However, the way the orientation function is obtained makes the starting point very significant. In fact, depending on the point of the contour we select to start, a different result may be generated. These results will differ in a certain shift and a constant, 2π, which is added to those values which appear on the opposite side of the signal when it is cyclically shifted. If we wish to identify a pattern without taking into account the first point we select, we must previously shift the resulting function to be able to compare it with the reference pattern and study how such constant affects the final result. In order to extract the shift that should be used, we could locate the point where correlation is maximum. Nevertheless, this requires a high computational cost and correlation may produce false identifications since it may be quite high for orientation functions corresponding to largely different shapes. As shown in figures 5.2 and 5.3, and in equation 5.1, the correlation for the orientation functions of a circle and a square is higher than 0.98, which leaves a very narrow range to set a threshold between similar and different shapes.

$$C(f_1, f_2) = \frac{\sum\limits_{i=0}^{L-1} f_i^1 f_i^2}{\|f^1\|\,\|f^2\|} = 0.984\,12 \tag{5.1}$$

That is why we use Fourier coefficients to reduce the number of computations to perform and, at the same time, estimate the most convenient shift to apply. These coefficients will provide not only the best shift to compare the signals, but also a discrimination function to determine how similar they are. In the pioneering work by Zahn and Roskies [ZR72], the optimum criteria as well as the consequences of certain transformations on Fourier coefficients are studied in the case of polygonal shapes

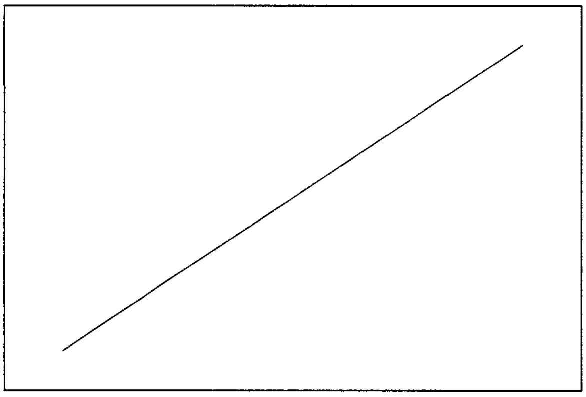


Figure 5.2: Orientation function for a circle.

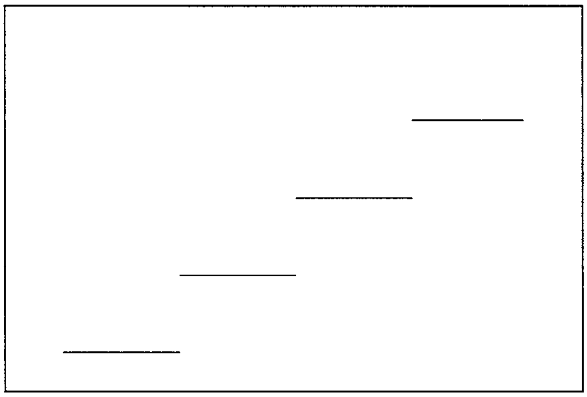


Figure 5.3: Orientation function for a square.

using continuous Fourier series. In this work, we generalize the results presented by Zahn and Roskies [ZR72] to the case of uniformly distributed sample sets of points of the boundary and we use the Fourier Transform for the characterization. More recently, Rui et al. [RSH96] describe a set of modified Fourier descriptors for shape representation. The authors use a uniformly sampled set of points of the boundary and a similarity distance based on a combination of the magnitude and phase of the Fourier coefficients. The way we extract the boundary points as well as the discrimination function that we propose differ from those presented by Zahn and Roskies [ZR72] and Rui et al. [RSH96].

Let f_n be a one-dimensional discrete signal corresponding to the orientation function of an object and consisting of L values, its discrete Fourier transform coefficients can be obtained as:

$$\widehat{f_k} = \tfrac{1}{L}\textstyle\sum_{n=0}^{L-1} f_n e^{-i\frac{2\pi kn}{L}} \qquad \forall\ k = 0,1,2,...,L-1$$

We are interested in identifying shapes regardless of their sizes. To be able to compare two shapes without taking into account their sizes, the sequences obtained for orientation functions are normalized in their length, in such a way that all of them are equally long, and thus avoiding the generation of quite different results for similar shapes of variable size. Therefore, the variations in the orientation functions are represented with respect to the total length of the contour, and not according to the real physical size of the object. This requires an interpolation of the values in the orientation function, since the original positions must be transformed to a different length which may not be one of its divisors or multiples, and this will transform the initially integer indices into real coordinates. Moreover, the length we use is always a power of 2, in order to use fast Fourier transform and reduce the computational cost.

When the object is rotated a certain angle θ, all points on the contour will undergo an increase in the values of their respective orientations. However, this increase will be the same for all of them, provided it is a solid whose shape is not altered by the rotation. This phenomenon only affects order 0 coefficient of the orientation function and not any other. Let g_n be the orientation signature of the object described by f_n after a rotation of a certain angle θ, its coefficients are:

$$\widehat{g_0} = \frac{1}{L}\sum_{n=0}^{L-1} g_n = \frac{1}{L}\sum_{n=0}^{L-1}(f_n + \theta) = \theta + \frac{1}{L}\sum_{n=0}^{L-1} f_n = \widehat{f_0} + \theta$$

$$\widehat{g_k} = \tfrac{1}{L}\textstyle\sum_{n=0}^{L-1} g_n e^{-i\frac{2\pi kn}{L}} = \tfrac{1}{L}\sum_{n=0}^{L-1}(f_n + \theta)e^{-i\frac{2\pi kn}{L}} = \tfrac{1}{L}\sum_{n=0}^{L-1} f_n e^{-i\frac{2\pi kn}{L}} = \widehat{f_k} \qquad \forall k \neq 0$$

Once we have seen that the parameters we are using are invariant to changes in the size of the object as well as to rotations, provided we do not use order 0 coefficients, the following section shows how the right shift can be extracted from the relationships between Fourier coefficients of signals which are obtained from a different starting point.

5.2 Influence of the starting point of the orientation sequence

To extract the sequence corresponding to a closed curve, any point on the contour can be used as start. However, we must take into account the consequences described in the previous chapter. Firstly, a shift is observed in the values which constitute the signal. Secondly, the exigency of continuity causes an increase in the values which appear on the opposite side of the sequence. These two effects are shown in figures 5.9 and 5.10. We will show some results concerning the influence of the starting point of the orientation sequence for continuous and discrete signals.

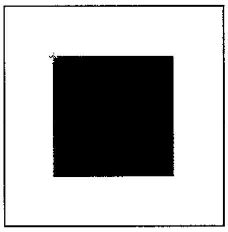


Figure 5.4: Square.

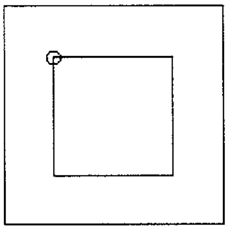


Figure 5.5: Contour for the square in figure 5.4 and starting point 1.

5.2.1 Continuous signals

If $g(.)$ and $f(.)$ are continuous signals and $g(.)$ is a shifted and corrected version of $f(.)$, according to the transformation described above, the relationship between their respective Fourier transforms can be used to extract the shift produced by starting

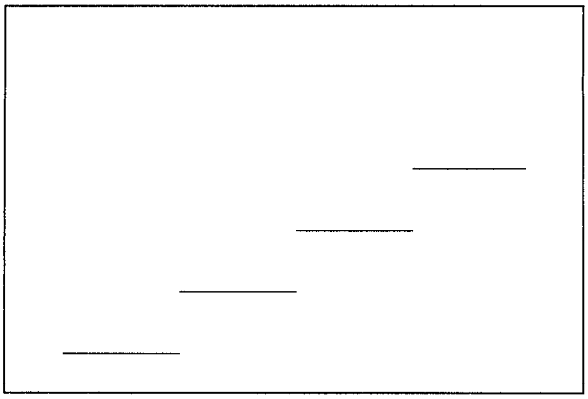


Figure 5.6: Orientation function for the square starting at the point in figure 5.5.

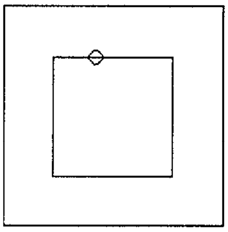


Figure 5.7: Contour for the square in figure 5.4 and starting point 2.

the contour at a different point. Once the shift has been extracted, the functions can be directly compared. Next, we will study the influence of such a transformation on the one-dimensional signal which constitutes the orientation function. Firstly, we use function h_a, described in equation (5.2), to add the constant 2π to a part of the signal.

$$h_a(x) = \begin{cases} 1 & if \quad 0 < x < a \\ 0 & if \quad a < x < L \end{cases} \tag{5.2}$$

Afterwards, we consider the relationship between the Fourier transforms of $g(.)$ and $f(.)$, which will lead us to an expression for extracting the shift, as shown below, where $w = 2\pi/L$:

$$g(x) = f_a(x) = f(x+a) + 2\pi h_a(x+a)$$

$$g(x-a) = f(x) + 2\pi h_a(x)$$

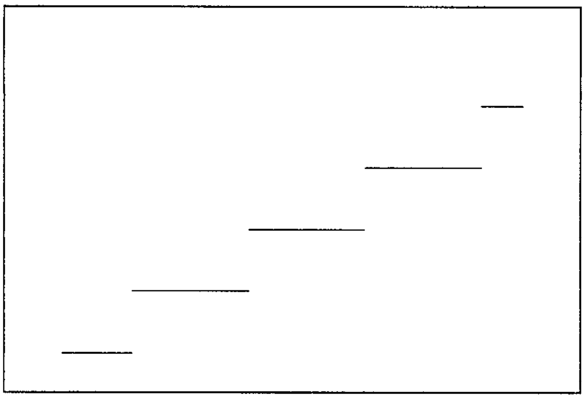


Figure 5.8: Orientation function for the square starting at the point in figure 5.7.

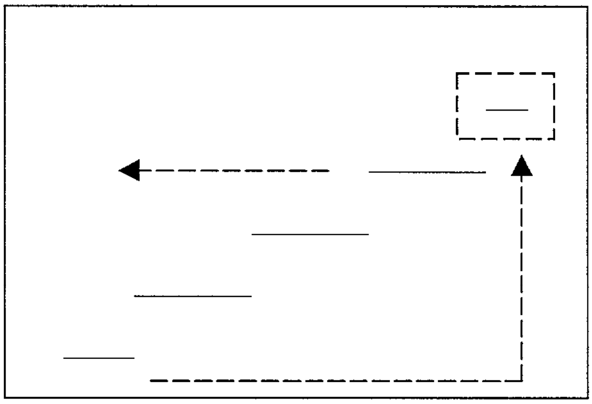


Figure 5.9: Differences between functions in figures 5.6 and 5.8.

$$g'(x-a) = f'(x) + 2\pi\left[\delta(x) - \delta(x-a)\right]$$

$$e^{-iwa} iw\widehat{g}(w) = iw\widehat{f}(w) + 2\pi\left[1 - e^{-iwa}\right]$$

$$e^{-iwa}\left[2\pi + iw\widehat{g}(w)\right] = 2\pi + iw\widehat{f}(w)$$

$$e^{-iwa} = \frac{2\pi + iw\widehat{f}(w)}{2\pi + iw\widehat{g}(w)}$$

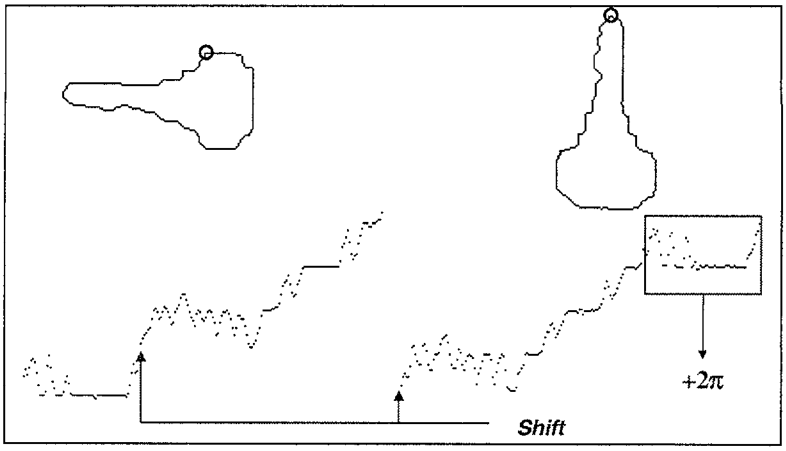


Figure 5.10: Effects of starting the sequence at a different point of the contour.

5.2.2 Discrete signals

The expressions in the previous section would allow us to identify the shift which relates two contours extracted from the same shape if they where continuous signals. Nevertheless, the relationships that have been extracted for continuous signals must be transformed if we are working with discrete signals. Let g_n be the signal resulting of shifting f_n a positions and increasing the values which appear on the opposite side in 2π, i.e. they correspond to the same shape starting at a different point of the contour, their coefficients are related as in equation (5.3).

$$g_n = f_n^a = \begin{cases} f_{n+a} & if \;\; n = 0, ..., L-a-1 \\ 2\pi + f_{n-(L-a)} & if \;\; n = L-a, ..., L-1 \end{cases} \tag{5.3}$$

This signal will generate the following coefficients:

$$
\begin{aligned}
\widehat{g}_k = \widehat{f^a_n} &= \frac{1}{L}\sum_{n=0}^{L-1} f^a_n e^{-i\frac{2\pi kn}{L}} \\
&= \frac{1}{L}\left(\sum_{n=0}^{L-a-1} f_{n+a} e^{-i\frac{2\pi kn}{L}} + \sum_{n=L-a}^{L-1} \left(2\pi + f_{n-(L-a)}\right) e^{-i\frac{2\pi kn}{L}}\right) \\
&= \frac{1}{L}\left(\sum_{m=a}^{L-1} f_m e^{-i\frac{2\pi k(m-a)}{L}} + \sum_{m=0}^{a-1} f_m e^{-i\frac{2\pi k(m-a+L)}{L}} + 2\pi \sum_{n=L-a}^{L-1} e^{-i\frac{2\pi kn}{L}}\right) \\
&= e^{i\frac{2\pi ka}{L}}\widehat{f}_k + \frac{2\pi\left(e^{-i\frac{2\pi k(L-a)}{L}} - e^{-i2\pi k}\right)}{L\left(1 - e^{-i\frac{2\pi k}{L}}\right)} \\
&= e^{i\frac{2\pi ka}{L}}\widehat{f}_k + \frac{2\pi\left(e^{i\frac{2\pi ka}{L}} - 1\right)}{L\left(1 - e^{-i\frac{2\pi k}{L}}\right)} \quad \forall\, k \neq 0
\end{aligned}
$$

From the previous equation, the following relationship between the coefficients of both signals can be extracted:

$$
\widehat{g}_k\left(1 - e^{-i\frac{2\pi k}{L}}\right) = e^{i\frac{2\pi ka}{L}}\widehat{f}_k\left(1 - e^{-i\frac{2\pi k}{L}}\right) + \frac{2\pi}{L}\left(e^{i\frac{2\pi ka}{L}} - 1\right)
$$

$$
e^{i\frac{2\pi ka}{L}} = \frac{\frac{2\pi}{L} + \widehat{g}_k\left(1 - e^{-i\frac{2\pi k}{L}}\right)}{\frac{2\pi}{L} + \widehat{f}_k\left(1 - e^{-i\frac{2\pi k}{L}}\right)}
$$

$$
e^{i\frac{2\pi ka}{L}} = \frac{2\pi + L\widehat{g}_k\left(1 - e^{-i\frac{2\pi k}{L}}\right)}{2\pi + L\widehat{f}_k\left(1 - e^{-i\frac{2\pi k}{L}}\right)} \quad \forall k = -\tfrac{L}{2}, ..., -1, 1, ..., \tfrac{L}{2} \tag{5.4}
$$

Taking into account that order 0 coefficient is related to the mean value of the signal and is altered by a rotation of the figure, it is not suitable for our purposes. On the other hand, the larger the order of the coefficient, the more sensitive it is to noise. If we only consider the relationship between order k coefficients of both signals ($k \neq 0$), we can estimate the shift as in equations (5.5) or (5.6).

$$
e^{i\frac{2\pi ak}{L}} = \frac{2\pi + L\widehat{g}_k\left(1 - e^{-i\frac{2\pi k}{L}}\right)}{2\pi + L\widehat{f}_k\left(1 - e^{-i\frac{2\pi k}{L}}\right)} \tag{5.5}
$$

$$a = -\frac{iL}{2\pi k} \ln \left(\frac{2\pi + L\widehat{g}_k \left(1 - e^{-i\frac{2\pi k}{L}}\right)}{2\pi + L\widehat{f}_k \left(1 - e^{-i\frac{2\pi k}{L}}\right)} \right) \tag{5.6}$$

When both series correspond to the same shape, the value obtained for the shift a must be real. However, if the objects are similar but not identical, the real component of this value will offer an estimation for the shift which should be used to compare both sequences and determine whether they represent the same shape. Since, in most cases, the fit of the reference sequence with the extracted one will not be perfect, the value obtained for a will provide us with a first approximation. The following sections describe how a more accurate value can be obtained.

5.3 Sequence direction and symmetrical shapes association

We have considered the problem of starting the orientation function of a closed curve at a different point, but next, we must see what happens if we choose the opposite direction to continue. If we go through the contour starting at the same point as in the reference pattern, but in the opposite direction, the sequence g_n that we will obtain is related to the original one f_n as described in equation (5.7).

$$g_n = \begin{cases} f_0 & if \quad n = 0 \\ f_{L-n} + 2\pi & if \quad n = 1, ..., L-1 \end{cases} \tag{5.7}$$

As a consequence, Fourier transform coefficients are modified in the following way:

$$\begin{aligned} \widehat{g}_k &= \frac{1}{L}\sum_{n=0}^{L-1} g_n e^{-i\frac{2\pi kn}{L}} \\ &= \frac{1}{L}\left(f_0 + \sum_{n=1}^{L-1}\left(f_{L-n} + 2\pi\right)e^{-i\frac{2\pi kn}{L}}\right) \\ &= \frac{1}{L}\left(f_0 + \sum_{n=1}^{L-1}\left(f_n + 2\pi\right)e^{i\frac{2\pi kn}{L}}\right) \\ &= \frac{1}{L}\left(-2\pi + \sum_{n=0}^{L-1}\left(f_n + 2\pi\right)e^{i\frac{2\pi kn}{L}}\right) \\ &= \begin{cases} \widehat{f}_0 + \frac{2\pi(L-1)}{L} & if \quad k = 0 \\ \widehat{f}_{-k} - \frac{2\pi}{L} & if \quad k \neq 0 \end{cases} \end{aligned}$$

As the signals whose coefficients are being calculated are real, the previous expression can be written as:

$$\widehat{g}_k = \begin{cases} \widehat{f}_0 + \frac{2\pi(L-1)}{L} & if \quad k = 0 \\ \widehat{f}_k^* - \frac{2\pi}{L} & if \quad k \neq 0 \end{cases} \tag{5.8}$$

This phenomenon can be detected because the difference between the first and the last points of the sequence will be positive for one of the signals and negative for the other. For closed curves, it is -2π if we progress clockwise and 2π if we do it counterclockwise.

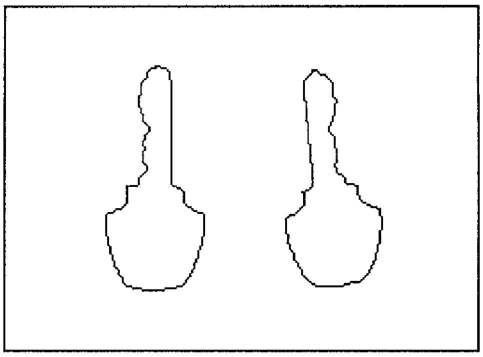


Figure 5.11: Contours of both sides of the same key.

If we are working with plane objects, e.g. we are trying to identify keys, they can be presented in two different forms, corresponding to both sides of the object, as shown in figure 5.11. However, one of them is a reflected version of the other and their orientation functions can be coupled if we consider the changes they will undergo. If g_n represents a shape that is a reflected version of that represented by f_n, starting at the same point, they can be related as shown in equation (5.9), where C is a value which depends on the starting point of the contour and the symmetry axis that has been used for reflection, but which remains constant for all points inside the sequence.

$$g_n = \begin{cases} C - f_0 & if \;\; n = 0 \\ C - f_{L-n} + 2\pi & if \;\; n = 1, ..., L-1 \end{cases} \tag{5.9}$$

An example of reflected shape representations is shown in figure 5.12.

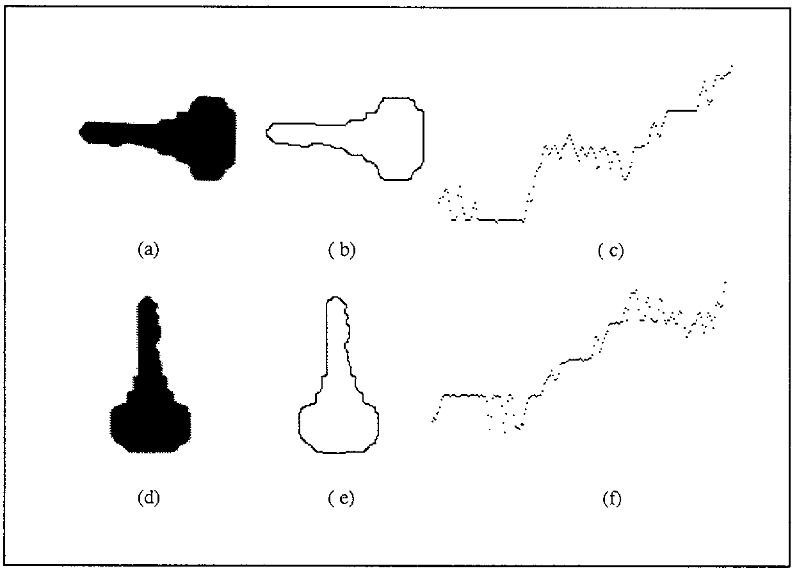


Figure 5.12: (a) and (d) Input images corresponding to the same key. (b) and (e) Contours. (c) and (f) Orientation functions.

This will modify Fourier transform coefficients as follows:

$$\begin{aligned}
\widehat{g}_k &= \frac{1}{L}\sum_{n=0}^{L-1} g_n e^{-i\frac{2\pi kn}{L}} \\
&= \frac{1}{L}\left(C - f_0 + \sum_{n=1}^{L-1}(C - f_{L-n} + 2\pi)\, e^{-i\frac{2\pi kn}{L}}\right) \\
&= \frac{1}{L}\left(C - f_0 + \sum_{n=1}^{L-1}(C - f_n + 2\pi)\, e^{i\frac{2\pi kn}{L}}\right) \\
&= \frac{1}{L}\left(-2\pi + \sum_{n=0}^{L-1}(C - f_n + 2\pi)\, e^{i\frac{2\pi kn}{L}}\right) \\
&= \begin{cases} -\widehat{f}_0 + \frac{2\pi(L-1)}{L} + C & if \quad k = 0 \\ -\widehat{f}_{-k} - \frac{2\pi}{L} & if \quad k \neq 0 \end{cases}
\end{aligned}$$

Taking into account the properties of Fourier coefficients, we can write this relationship as:

$$\widehat{g}_k = \begin{cases} -\widehat{f}_0 + \frac{2\pi(L-1)}{L} + C & if \quad k = 0 \\ -\widehat{f}^*_k - \frac{2\pi}{L} & if \quad k \neq 0 \end{cases} \tag{5.10}$$

Equations (5.8) and (5.10) will allow us to build energy functions for shape characterization, as described in the next section.

5.4 Shape characterization

Once we have studied how the functions we use for the description of a contour are affected by certain transformations, we consider now the comparison between two or more such functions.

5.4.1 Energy function

As said before, we can use any but order 0 coefficient, since it is related to the mean value, and it should not affect the result. We also stated that the higher the order of the coefficient, the more sensitive it is to noise. That may lead us to think that we should use lower order coefficients. However, when a shape has r-fold rotational symmetry and fits itself under a rotation of $2\pi/r$, those coefficients whose order is not multiple of r are null, thus avoiding to extract a right shift from them. For example, coefficients 1, 2 and 3 are null for a perfect square, and only those which are multiples of 4 provide some useful information for our purpose. In case we use several coefficients to estimate the shift, instead of extracting it from only one of them, we can build an energy function as a sum of the errors for every coefficient. This will provide an accurate value for the shift as well as a similarity measure to compare shapes. From equation (5.4), which can be rewritten as equation (5.11), we can determine how good the relationship is for a certain value of a and a given coefficient order k, as shown in equation (5.12):

$$\widehat{g}_k = e^{i\frac{2\pi ka}{L}}\widehat{f}_k + \frac{2\pi\left(e^{i\frac{2\pi ka}{L}} - 1\right)}{L\left(1 - e^{-i\frac{2\pi k}{L}}\right)} \tag{5.11}$$

$$\begin{aligned} E_k(a) &= e^{i\frac{2\pi ka}{L}}\left(\widehat{f}_k + \frac{2\pi}{L\left(1 - e^{-i\frac{2\pi k}{L}}\right)}\right) - \left(\widehat{g}_k + \frac{2\pi}{L\left(1 - e^{-i\frac{2\pi k}{L}}\right)}\right) \\ &= e^{i\frac{2\pi ka}{L}}\widetilde{f}_k - \widetilde{g}_k \end{aligned} \tag{5.12}$$

$$\text{where } \widetilde{f}_k = \widehat{f}_k + \frac{2\pi}{L\left(1 - e^{-i\frac{2\pi k}{L}}\right)}$$

If we add the terms corresponding to every coefficient with non-null index, multiplying each one of them by its respective conjugate, we obtain the following function, where $\widetilde{f_k}$ and $\widetilde{g}_k$ are obtained from Fourier coefficients as shown above:

$$E(a) = \sum_{k=1}^{\frac{L}{2}} \left(e^{i\frac{2\pi ka}{L}}\widetilde{f_k} - \widetilde{g}_k\right)\left(e^{i\frac{2\pi ka}{L}}\widetilde{f_k} - \widetilde{g}_k\right)^* \tag{5.13}$$

$$= \sum_{k=1}^{\frac{L}{2}} \left(\left|\widetilde{f_k}\right|^2 + |\widetilde{g}_k|^2 - e^{i\frac{2\pi ka}{L}}\widetilde{f_k}\widetilde{g}_k^* - \left(e^{i\frac{2\pi ka}{L}}\widetilde{f_k}\widetilde{g}_k^*\right)^*\right)$$

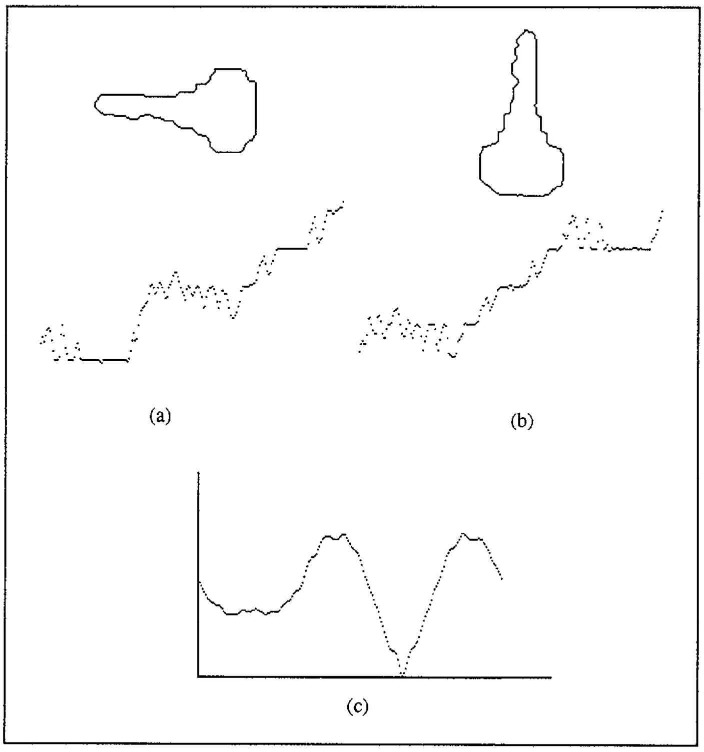


Figure 5.13: (a) and (b) Contours and orientation functions of two images corresponding to the same key. (c) Energy function $E(a)$ for shift estimation.

And now, we should extract the value for a which minimizes this function. Equation (5.14) shows the function whose zero-crossings are to be found for the minimization.

$$\frac{dE(a)}{da} = 0 \Longrightarrow \sum_{k=1}^{\frac{L}{2}} \left(\frac{2\pi i}{L}k\left(\widetilde{f_k}\widetilde{g}_k^* e^{i\frac{2\pi ka}{L}} - \widetilde{f}_k^*\widetilde{g}_k e^{-i\frac{2\pi ka}{L}}\right)\right) = 0 \tag{5.14}$$

$$\sum_{k=1}^{\frac{L}{2}} \left(k\left(\widetilde{f_k}\widetilde{g}_k^* e^{i\frac{2\pi ka}{L}} - \widetilde{f}_k^*\widetilde{g}_k e^{-i\frac{2\pi ka}{L}}\right)\right) = 0$$

$$\sum_{k=1}^{\frac{L}{2}} \left(k\left(\widetilde{f}_k\widetilde{g}_k^{*}z^{k} - \widetilde{f}_k^{*}\widetilde{g}_k z^{-k}\right)\right) = 0$$

$$\sum_{\substack{k=-\frac{L}{2}\\ k\neq 0}}^{\frac{L}{2}} \left(k\widetilde{f}_k\widetilde{g}_k^{*}z^{k}\right) = 0$$

$$\text{where } z = e^{i\frac{2\pi a}{L}}$$

Figure 5.13 shows an example with two images of the same key and the energy function for the corresponding shape representations. If we have obtained an approximation for a from one of the coefficients, we could use this value for an iterative scheme, such as Newton-Raphson method, to extract a more accurate one, as shown in equation (5.16). We use function $M(a)$ in equation (5.15) as a simplified expression of $dE(a)/da$, whose zero-crossings must be found.

$$M(a) = \sum_{k=1}^{\frac{L}{2}} \left(k\left(\widetilde{f}_k\widetilde{g}_k^{*}e^{i\frac{2\pi ka}{L}} - \widetilde{f}_k^{*}\widetilde{g}_k e^{-i\frac{2\pi ka}{L}}\right)\right) \tag{5.15}$$

$$\frac{dM(a)}{da} = \frac{2\pi i}{L}\sum_{k=1}^{\frac{L}{2}} \left(k^{2}\left(\widetilde{f}_k\widetilde{g}_k^{*}e^{i\frac{2\pi ka}{L}} + \widetilde{f}_k^{*}\widetilde{g}_k e^{-i\frac{2\pi ka}{L}}\right)\right)$$

$$\begin{aligned} a_{n+1} &= a_n - \frac{M(a_n)}{\frac{dM(a_n)}{da}} \\ &= a_n - \frac{\sum_{k=1}^{\frac{L}{2}}\left(k\left(\widetilde{f}_k\widetilde{g}_k^{*}e^{i\frac{2\pi ka_n}{L}} - \widetilde{f}_k^{*}\widetilde{g}_k e^{-i\frac{2\pi ka_n}{L}}\right)\right)}{\frac{2\pi i}{L}\sum_{k=1}^{\frac{L}{2}}\left(k^{2}\left(\widetilde{f}_k\widetilde{g}_k^{*}e^{i\frac{2\pi ka_n}{L}} + \widetilde{f}_k^{*}\widetilde{g}_k e^{-i\frac{2\pi ka_n}{L}}\right)\right)} \\ &= a_n - \frac{L\sum_{k=1}^{\frac{L}{2}}\left(k\,\mathrm{Im}\left(\widetilde{f}_k\widetilde{g}_k^{*}e^{i\frac{2\pi ka_n}{L}}\right)\right)}{2\pi\sum_{k=1}^{\frac{L}{2}}\left(k^{2}\,\mathrm{Re}\left(\widetilde{f}_k\widetilde{g}_k^{*}e^{i\frac{2\pi ka_n}{L}}\right)\right)} \end{aligned} \tag{5.16}$$

Due to the fact that this method converges quickly towards a minimum when we are close to it but may converge to a maximum if we are not close enough, we use a Levenberg-Marquardt scheme as shown below:

$$a_{n+1} = a_n - \frac{\sum_{k=1}^{\frac{L}{2}} \left(k \left(\widetilde{f}_k \widetilde{g}_k^* e^{i\frac{2\pi k a_n}{L}} - \widetilde{f}_k^* \widetilde{g}_k e^{-i\frac{2\pi k a_n}{L}} \right) \right)}{\lambda + \frac{2\pi i}{L} \sum_{k=1}^{\frac{L}{2}} \left(k^2 \left(\widetilde{f}_k \widetilde{g}_k^* e^{i\frac{2\pi k a_n}{L}} + \widetilde{f}_k^* \widetilde{g}_k e^{-i\frac{2\pi k a_n}{L}} \right) \right)}$$

When it provides a better approximation, i.e. a point where the energy has been reduced, we consider the resulting value as the new estimation, decreasing the value of λ and thus tending to Newton-Raphson method. Otherwise, we refuse the new value, increasing the value of λ to move oppositely to the first derivative.

Once we have found a measure of the similarity of two contours, a threshold must be set to decide whether they come from the same shape or not and, in case they do not, to determine how different they are. This value will depend on the practical application we deal with, but in order to standardize the energy values, a normalization process is carried out for a given set of shapes. For normalization, we first calculate the average of the energy values obtained when comparing two different images corresponding to the same key. Afterwards, we divide all the values in the table by this factor. In table 5.1, the normalized minimum energy values are shown for the comparisons of 9 different images corresponding to 3 keys (see figure 5.14). In this case, equation (5.13) has been used and $k_{n:m}$ corresponds to the m^{th} image of the n^{th} key (n^{th} key of the m^{th} row). Normalized energy values around or lower than 1 indicate a great similarity between the shapes which are compared, while those values which are much higher than 1 indicate that the contours correspond to clearly different shapes. When comparing the keys in the last row of figure 5.14 with those in the first two rows, the relationship described for reflected shapes is used. In order to decide whether we must use the direct or the reflected relationship, we compare the results for both situations and determine whether the keys are presented in the same or the reflected position, according to the option which generates a lower value.

5.4.2 Weighting functions for frequency terms

Taking into account that the higher the order of the coefficient, the more sensitive it is to noise, we can weight the energy factors in equation (5.13) in such a way that the first coefficients are more significant than the last ones, as shown with the weighting function $w(.)$ in equation (5.17).

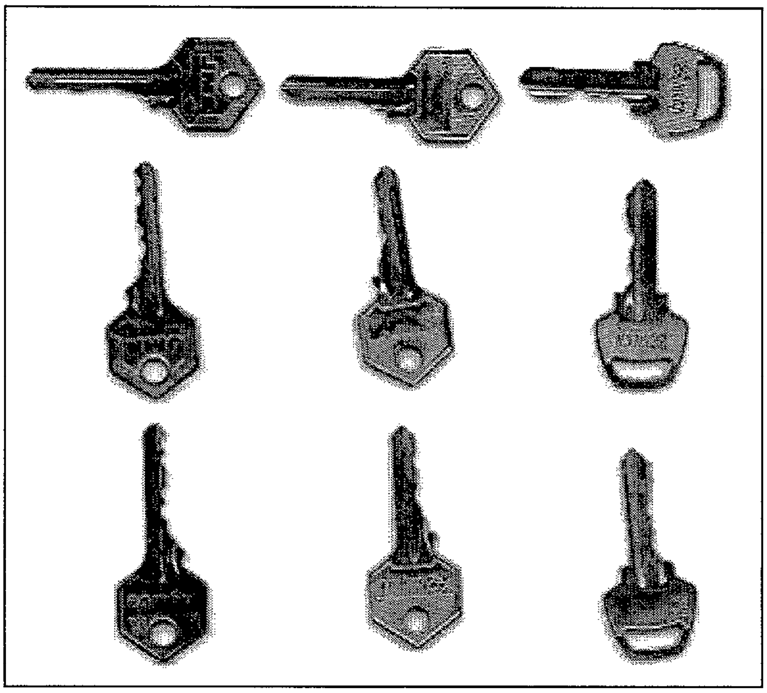


Figure 5.14: Images of three different keys in different positions and showing both sides.

$$E'(a) = \sum_{k=1}^{\frac{L}{2}} w\left(\frac{2k}{L}\right)\left(\left|\widetilde{f}_k\right|^2 + |\widetilde{g}_k|^2 - e^{i\frac{2\pi ka}{L}}\widetilde{f}_k\widetilde{g}_k^* - \left(e^{i\frac{2\pi ka}{L}}\widetilde{f}_k\widetilde{g}_k^*\right)^*\right) \tag{5.17}$$

With this new expression, those alterations of the signal due to the presence of noise, which affect more strongly higher order coefficients, are not so significant. When we compare two images of the same shape, the differences are due to the digitization process, the noise which has been introduced and the fact that we are working with estimations of the different parameters. However, when the images come from different shapes, the higher energy values are due to the dissimilarities between their contours. In the first case, the weighted version of the energy function will reduce the higher factors, thus decreasing the undesired effects. In the latter case, even if higher order factors are reduced, lower order ones, which carry the information on lower frequencies, are considerably maintained.

Different weighting functions have been tested to reduce higher frequency factors while preserving the information contained in lower frequencies. In order to test how suitable a certain weighting function is for a given set of shapes, we have divided

E_{min}	$k_{1:1}$	$k_{1:2}$	$k_{1:3}$	$k_{2:1}$	$k_{2:2}$	$k_{2:3}$	$k_{3:1}$	$k_{3:2}$	$k_{3:3}$
$k_{1:1}$	0.0000	1.1273	0.8219	3.5856	4.1306	4.1161	6.8020	5.9809	5.6455
$k_{1:2}$		0.0000	0.5607	5.4166	5.7934	5.7902	7.5247	6.4634	6.3026
$k_{1:3}$			0.0000	4.5853	5.2063	5.0216	7.4232	6.1860	6.0272
$k_{2:1}$				0.0000	0.9050	1.1134	8.6651	8.5584	8.0708
$k_{2:2}$					0.0000	1.0617	8.9780	6.7763	6.6304
$k_{2:3}$						0.0000	8.8944	8.9579	8.3986
$k_{3:1}$							0.0000	1.3734	1.2102
$k_{3:2}$								0.0000	0.8266
$k_{3:3}$									0.0000

Table 5.1: Normalized minimum energy values for keys in figure 5.14.

the results for our test set into two subsets. The first one, S, contains the results of the comparisons of images of the same key. The second one, D, contains the results of the comparisons of images of different keys. The parameter to test the quality of the weighting function, Q_w, is the ratio between the lowest energy obtained for two images of different shapes, $\min(D)$, and the highest energy for two images of the same shape, $\max(S)$. As far as $\min(D)$ is higher than $\max(S)$, a certain threshold can be set between both values in order to discriminate, but the higher the quotient Q_w, the clearer the distinction. When the discrimination threshold is very close to $\max(S)$, the probability of characterizing two images of the same shape as different will be high, but when it is very close to $\min(D)$, the probability of classifying two different shapes as equivalent will be increased. Hence the importance of achieving a high quotient.

$$Q_w = \frac{\min(D)}{\max(S)} \tag{5.18}$$

The weighting functions should be monotonic decreasing and non-negative in the interval $[0, 1]$. We also force the function to satisfy the condition $w(0) = 1$. When we use linear functions as those in figures 5.16 and 5.17, the results are improved with respect to the non-weighted energy function, as shown in table 5.2. We have also tested the results when the weighting function is quadratic, as in figures 5.18 and 5.19. Forcing the constraints $w(0) = 1$, $w(1) = 0$ and $w([0, 1]) \subseteq [0, 1]$, the maximum and minimum values for $w(0.5)$ are 0.75 and 0.25, respectively. These limits yield the functions in table A.7. As observed, the results of the second function are even better than those obtained with linear functions.

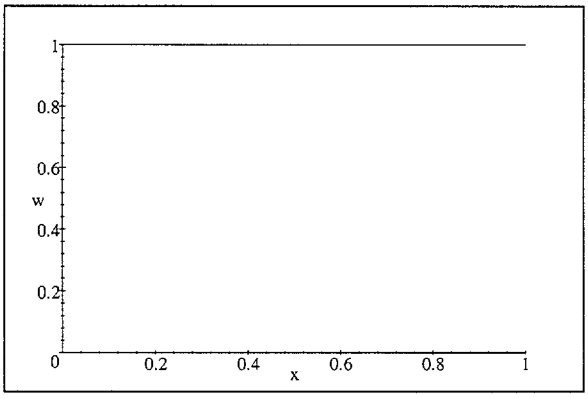


Figure 5.15: Constant function 1.

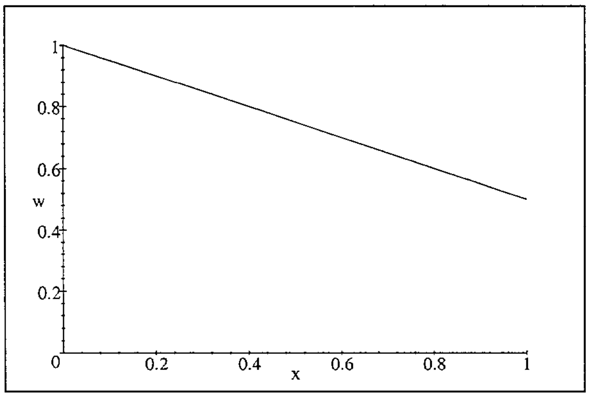


Figure 5.16: Linear function $1 - x/2$.

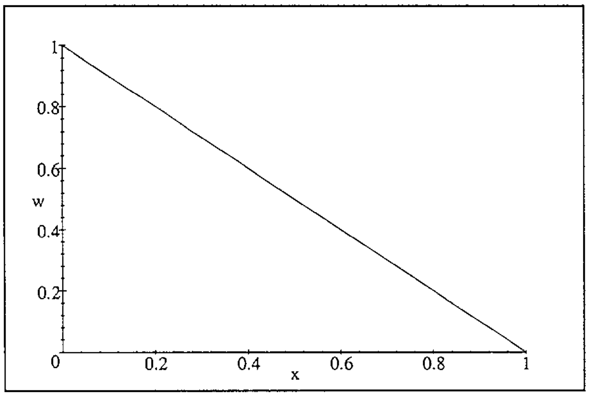


Figure 5.17: Linear function $1 - x$.

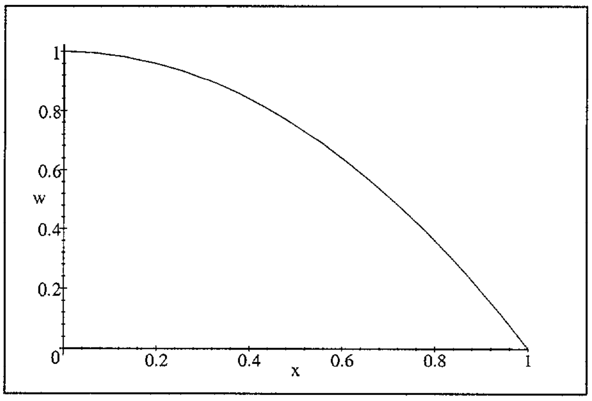


Figure 5.18: Quadratic function $-x^2 + 1$.

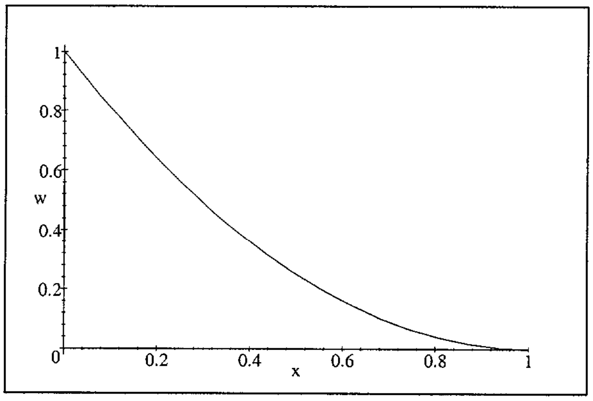


Figure 5.19: Quadratic function $x^2 - 2x + 1$.

$w(x)$	$\min(D)$	$\max(S)$	Q_w
1	3.5856	1.3734	2.6107
$1-\frac{x}{2}$	3.7549	1.3957	2.6903
$1-x$	3.9771	1.4250	2.7909

Table 5.2: Discrimination ratio for constant and linear functions.

$w(x)$	$\min(D)$	$\max(S)$	Q_w
$-x^2+1$	3.7838	1.3972	2.7081
x^2-2x+1	4.2645	1.4663	2.9083

Table 5.3: Discrimination ratio for quadratic functions.

However, the best ratios are observed when exponential functions of the form e^{-sx} are used, taking into account that the factor s in the exponent affects considerably the results. By varying the exponential factor s in the weighting function e^{-sx}, we obtain the ratios in table 5.4. Some examples of these functions are shown in figures 5.20, 5.21, 5.22 and 5.23. As observed, the ratios are improved as we increase the factor up to a maximum which is found around $s = 10$ (see figure 5.22). From this value on, the ratio decreases and it is lower than 1 from $s = 50$ on, which prevents us from using them as weighting functions for discrimination. This means that those exponential functions with a higher value of the parameter s reduce the lower frequency factors too much, thus eliminating the most relevant information about the shape.

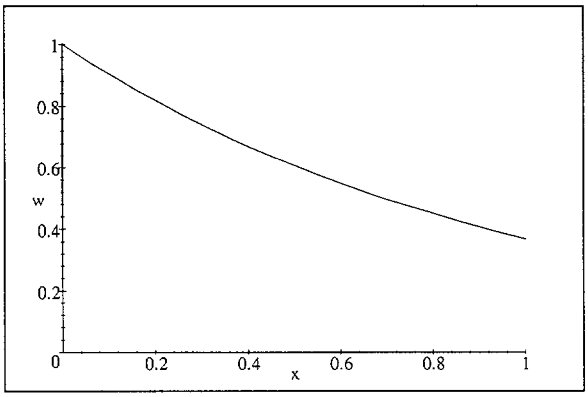


Figure 5.20: Exponential weighting function e^{-sx} for $s = 1$.

In figure 5.24 we can see the influence of the parameter s of the exponential

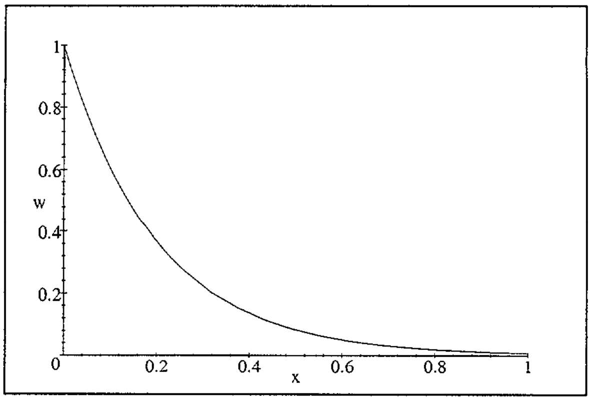


Figure 5.21: Exponential weighting function e^{-sx} for $s = 5$.

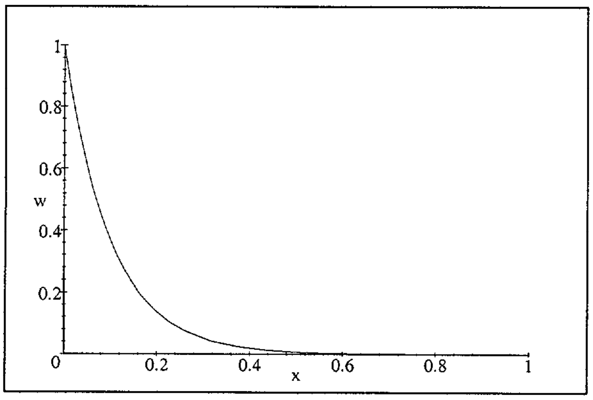


Figure 5.22: Exponential weighting function e^{-sx} for $s = 10$.

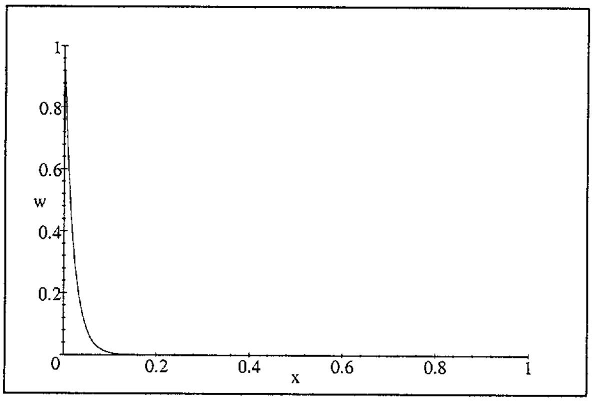


Figure 5.23: Exponential weighting function e^{-sx} for $s = 50$.

s	$w(x)$	$\min(D)$	$\max(S)$	Q_w
0	1	3.5856	1.3734	2.6107
1	e^{-x}	3.8642	1.4110	2.7386
5	e^{-5x}	4.5941	1.5125	3.0374
10	e^{-10x}	4.8422	1.5318	3.1611
15	e^{-15x}	4.3416	1.4837	2.9662
25	e^{-25x}	3.2275	1.5928	2.0263
50	e^{-50x}	1.7315	1.8574	0.9322
100	e^{-100x}	0.8781	2.0743	0.4233

Table 5.4: Discrimination ratio for exponential functions.

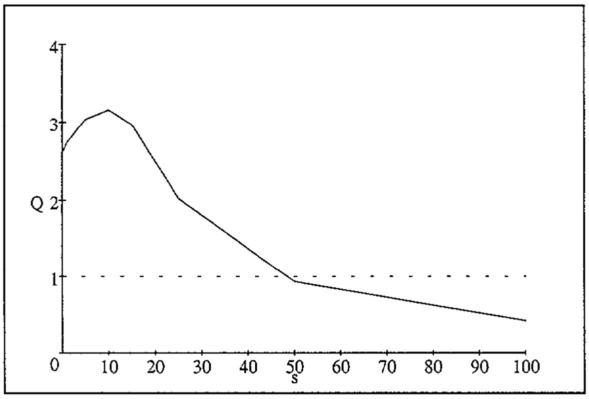


Figure 5.24: Discrimination ratio for different values of s between 0 and 100. Only those values over 1 allow setting a threshold to discriminate.

function on the discrimination ratio. Figure 5.25 shows a comparison for the ratios achieved with the different categories which have been tested. As observed, with exponential functions, the possibility of adjusting the parameter allows reaching higher quotients, which results in a better quality of the discrimination.

Table 5.5 shows the final values of normalized minimum energy for the keys in figure 5.14 when the factors are weighted as in equation (5.17), using $w(x) = e^{-10x}$ as weighting function. As observed when comparing tables 5.1 and 5.5, normalized energy values are not strongly altered when the images correspond to the same key, even if they present different orientations or a different side of the key is shown. However, energy values increase significantly for images of different keys when the weighting function is used, which results in a clearer discrimination of the shapes.

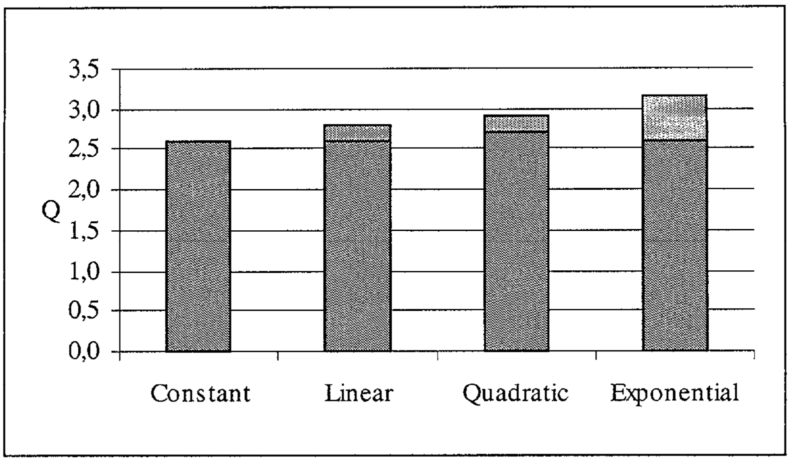


Figure 5.25: Results obtained for constant, linear, quadratic and exponential functions ($0 \leq s \leq 15$), used to weight the energy terms. Dark and light grey signal the worst and the best values for the discrimination ratio into each category, respectively.

This proves that not all terms in the energy function are equally significant for the discrimination of a certain set of forms. A reduction in the higher frequencies will also reduce the noise effects and facilitate the discrimination criteria. Nevertheless, the more detailed the forms, the higher the frequencies we will need to consider.

$E'_{\min}$	$k_{1:1}$	$k_{1:2}$	$k_{1:3}$	$k_{2:1}$	$k_{2:2}$	$k_{2:3}$	$k_{3:1}$	$k_{3:2}$	$k_{3:3}$
$k_{1:1}$	0.0000	1.1595	0.7754	4.8422	5.4602	4.8694	11.0473	9.0618	8.5987
$k_{1:2}$		0.0000	0.4007	8.2601	8.5063	7.0163	11.9956	9.2841	8.9991
$k_{1:3}$			0.0000	6.7835	6.8951	5.9262	11.8338	9.0418	8.7061
$k_{2:1}$				0.0000	0.8663	1.3553	14.1622	14.3641	13.0408
$k_{2:2}$					0.0000	1.0698	14.5646	12.3600	11.6192
$k_{2:3}$						0.0000	14.3818	15.8590	13.7275
$k_{3:1}$							0.0000	1.5318	1.1737
$k_{3:2}$								0.0000	0.6675
$k_{3:3}$									0.0000

Table 5.5: Normalized minimum energy values for keys in figure 5.14 and weighting function $w(x) = e^{-10x}$.

5.5 Shape-based image clustering

Shape characterization can be used in many different ways. This section shows an example of a database containing pictures of different kinds of marine animals and how they can be classified, in such a way that, for a given picture of a fish, the most similar ones are selected. This allows identifying and relating different species. These pictures are shown in figures 5.27-5.36.

This fish database has been kindly provided to us by Professor Farzin Mokhtarian at the Centre for Vision, Speech, and Signal Processing of the University of Surrey, United Kingdom [Mok01]. Of course, this classification is performed according to the shape of the silhouette and no other aspects, such as biological factors, are considered to determine which fishes are the most similar to the selected one.

Assuming that no other information can be obtained from the images of the fishes, the minimum energy value is calculated, as explained in the previous sections, for the coupling of the selected fish and every fish in the database. If the fish is contained in it, it should yield the minimum energy value and the other fishes could be ordered according to their respective energies, which reflect their similarity. This allows identifying an image of a fish as well as implementing categorizations of families. If a certain set of models for every family is built, they could be used as reference prototypes with the most significant elements of the family, even if it does not correspond to a member of the family.

Due to the nature of these images, certain parts of some of them are one-pixel wide. In order to prevent the process from being stalled because once these points are visited no other can be used to continue extracting the contour, a four-neighbor dilation algorithm is applied before starting, thus slightly widening the shapes, as shown in figure 5.26.

The following pages show de images in the database and examples in figures 5.37-5.44 show comparisons using the technique described in the previous sections. Among the 1000 images of marine animals contained in the database, one is selected, and the 10 best comparisons are shown. Of course, as the selected image is contained in the set, the best match corresponds to itself, and the energy factor is 0.

Taking into account that the shape of the contour is the only reference considered to associate the fishes, even if rotations, translations, scaling or reflections of the shapes do not alter the result, the comparison is affected by the curvature of the body and the tail, the shape and proportional size of the fins or the moustaches and

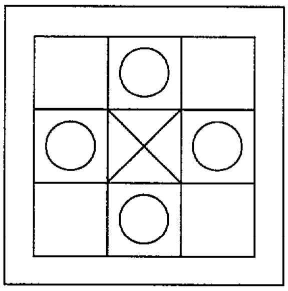


Figure 5.26: Four-neighbor dilation process.

small extremities that some of them present. Furthermore, the proportion of the contour which each part of it constitutes is a crucial factor.

In figure 5.45, we can see what the result would be for image 716 if we used the energy without the weighting function. When we compare it with the results in figure 5.44, we observe the improvement produced when the weighting function is considered.

When we compare the results for shape 779 with and without weighting function, shown in figures 5.46 and 5.47, even if the most similar shapes (780 and 616) produce the lowest values in both cases, the difference with quite different shapes is much stronger when the weighting function is used.

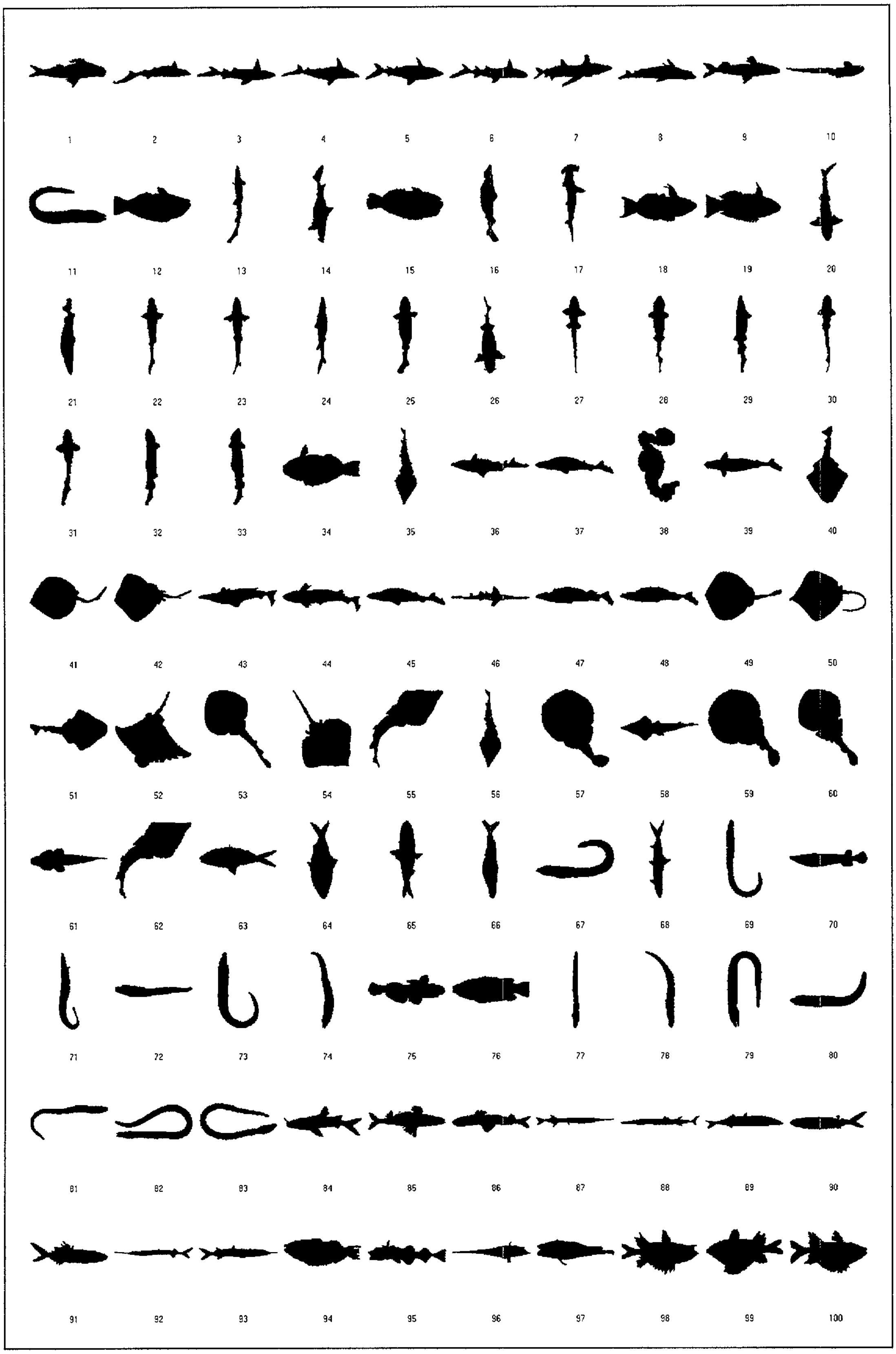


Figure 5.27: Database pictures (1).

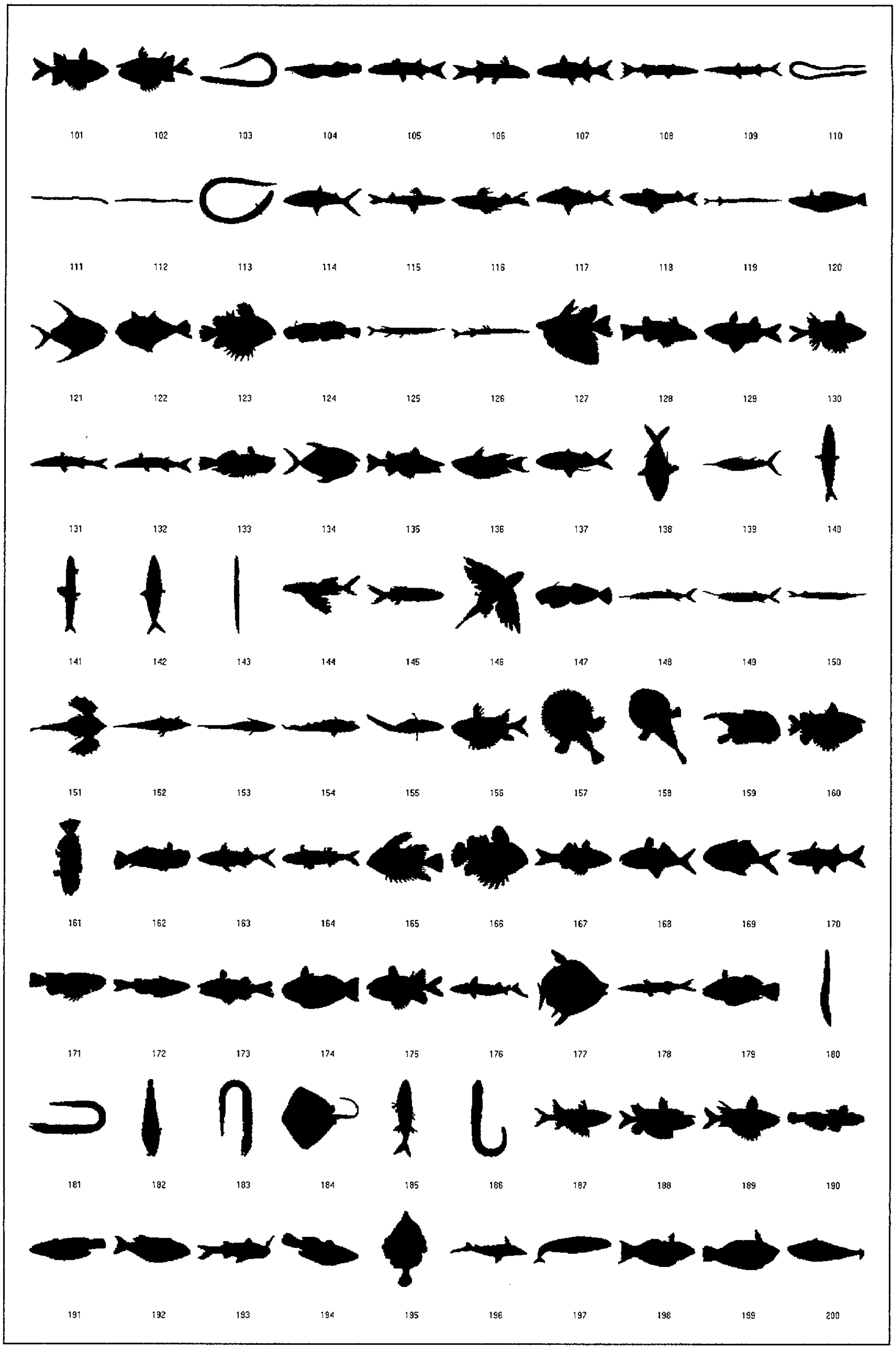


Figure 5.28: Database pictures (2).

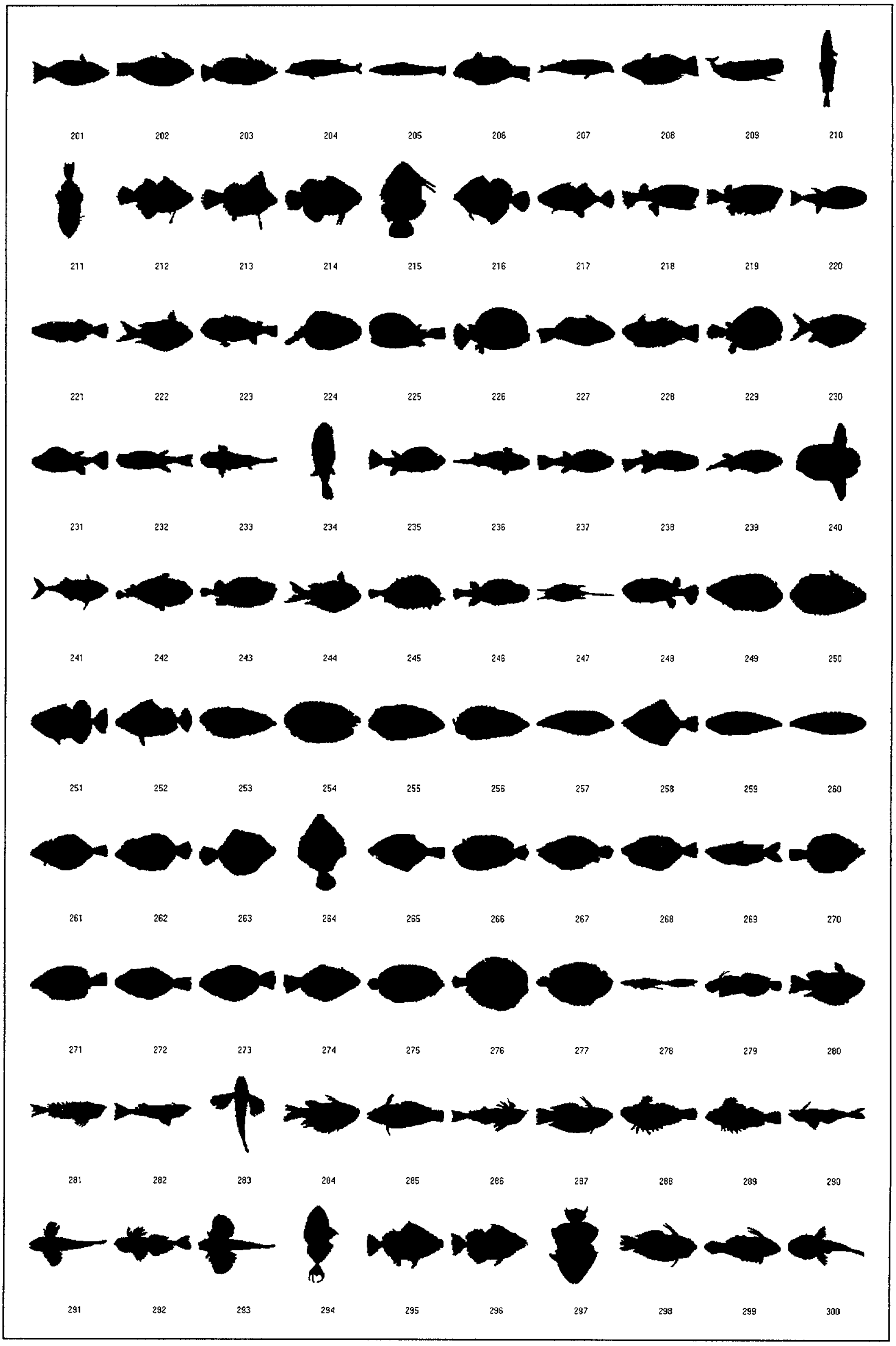


Figure 5.29: Database pictures (3).

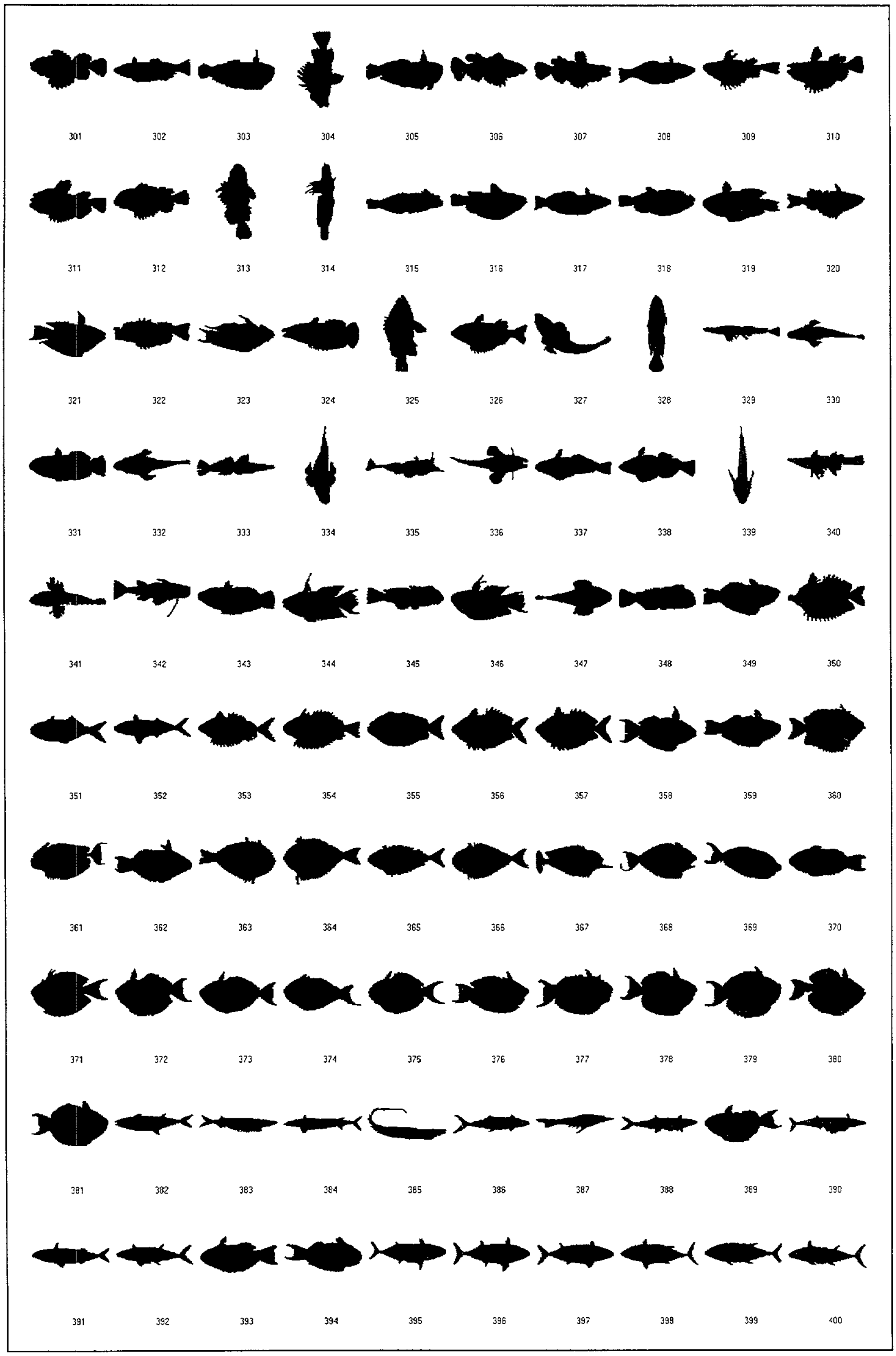


Figure 5.30: Database pictures (4).

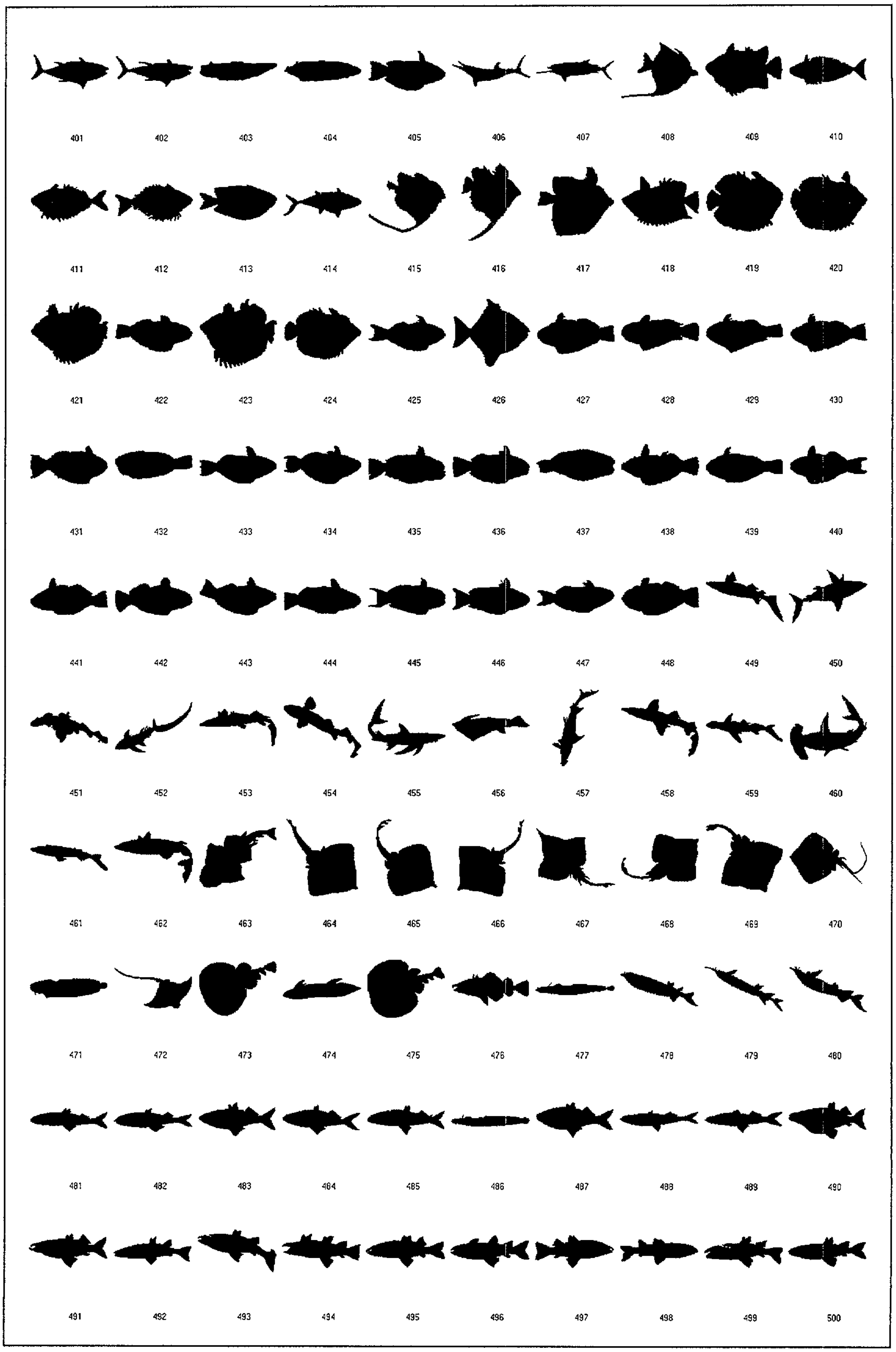


Figure 5.31: Database pictures (5).

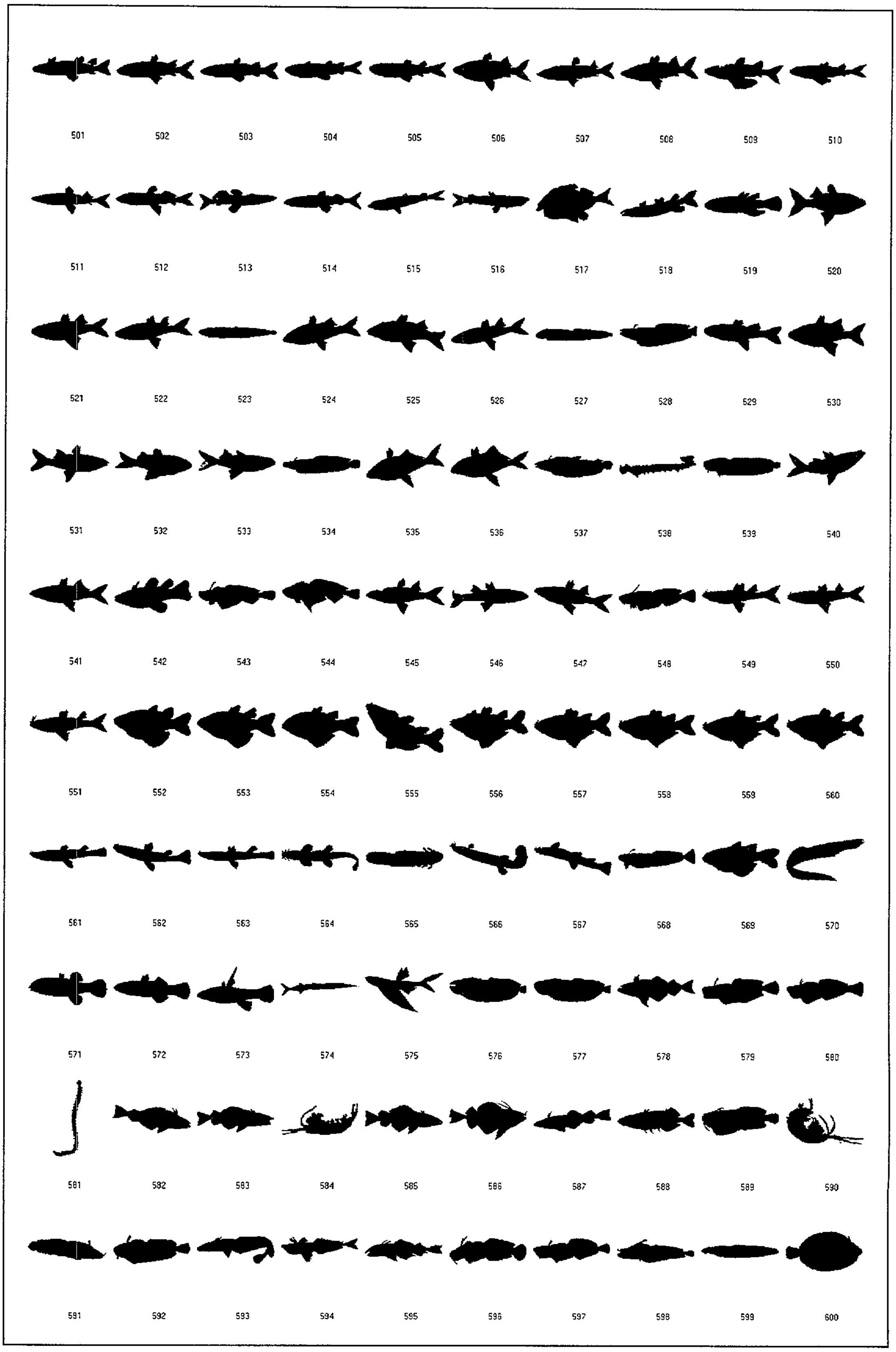


Figure 5.32: Database pictures (6).

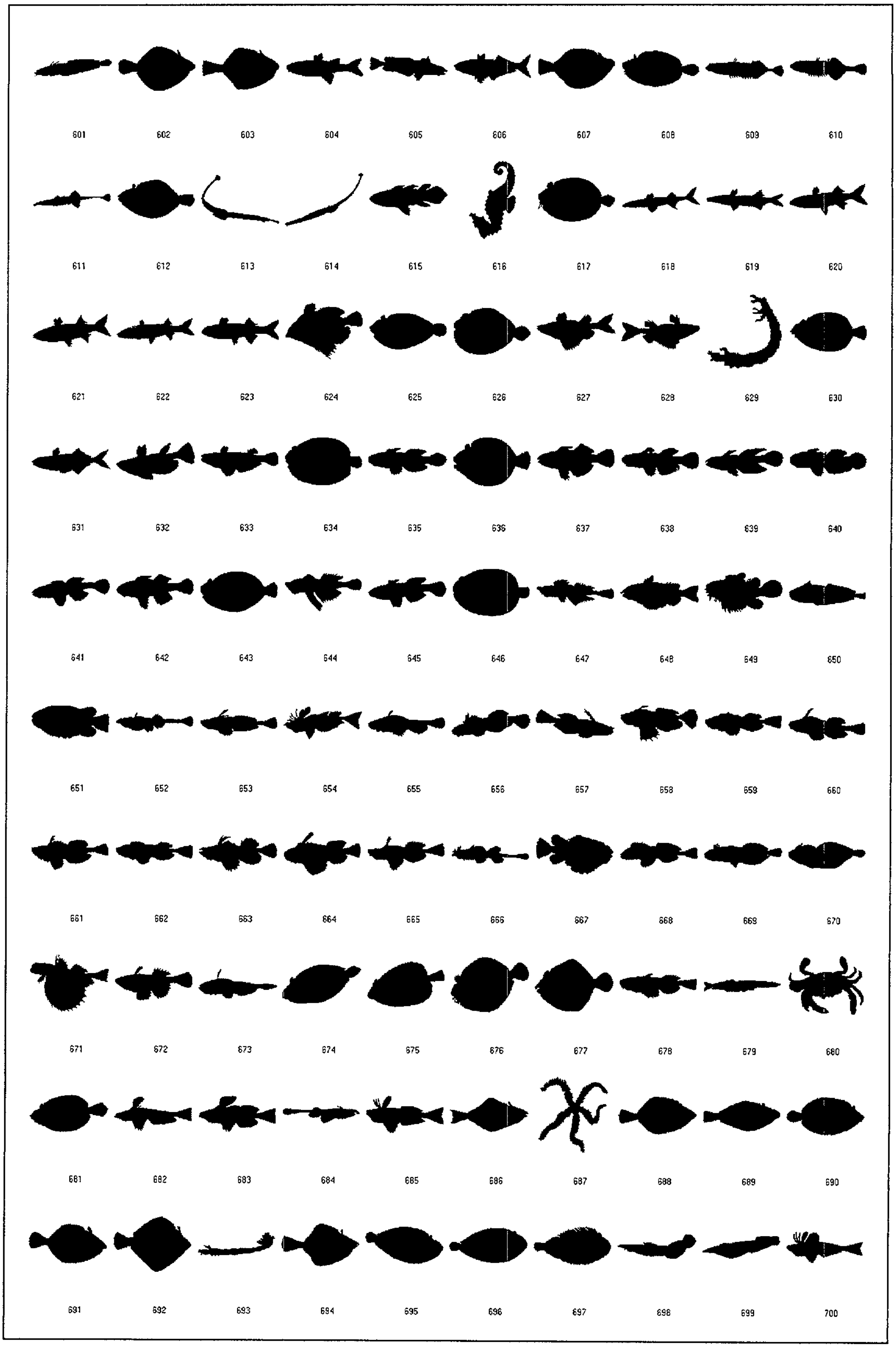


Figure 5.33: Database pictures (7).

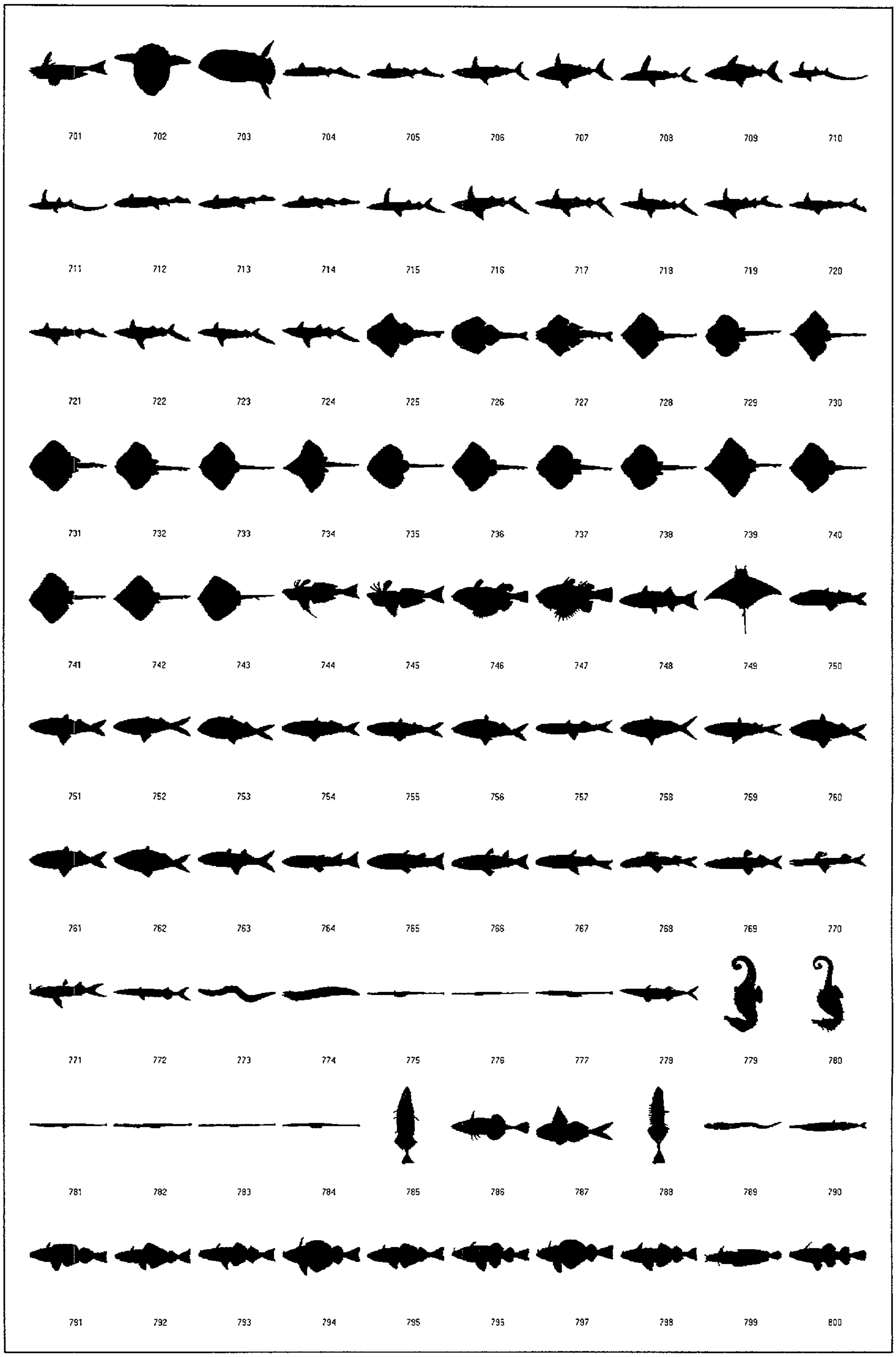


Figure 5.34: Database pictures (8).

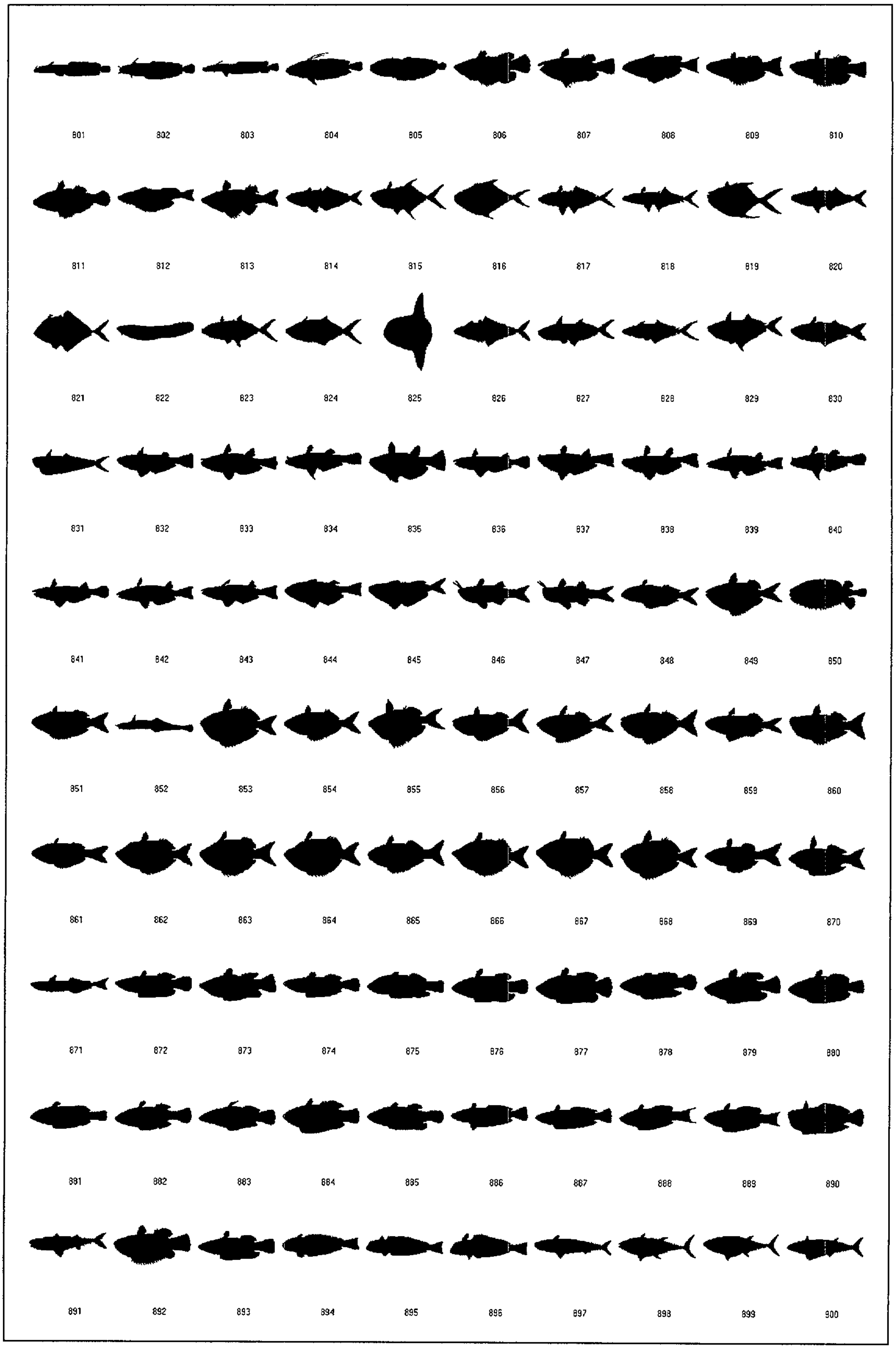


Figure 5.35: Database pictures (9).

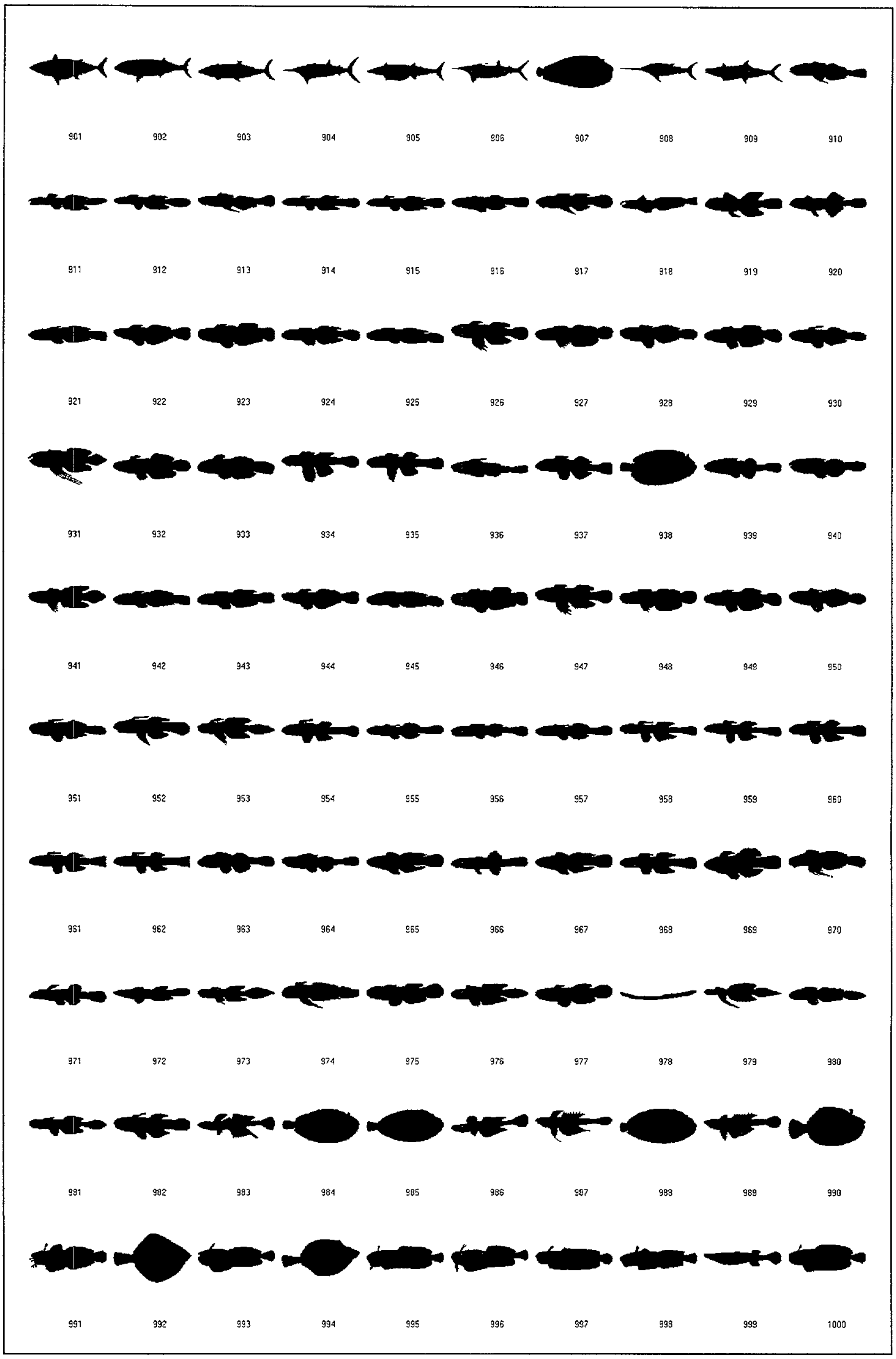


Figure 5.36: Database pictures (10).

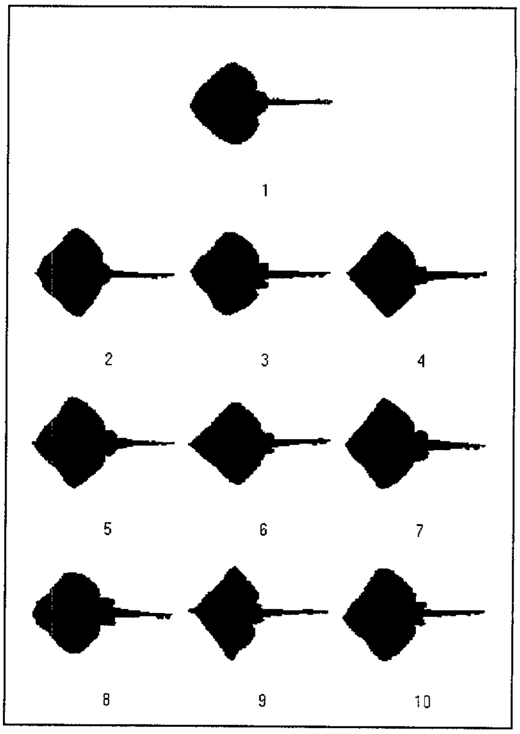


Figure 5.37: Example of the results of searching for similar shapes for shape 735.

order	shape number	weighted energy
1	735	0.00
2	733	717.21
3	738	837.46
4	728	916.66
5	740	1390.22
6	742	1401.43
7	736	1463.12
8	737	1573.66
9	730	1634.46
10	741	1660.60

Table 5.6: Lowest weighted energy values for shape 735.

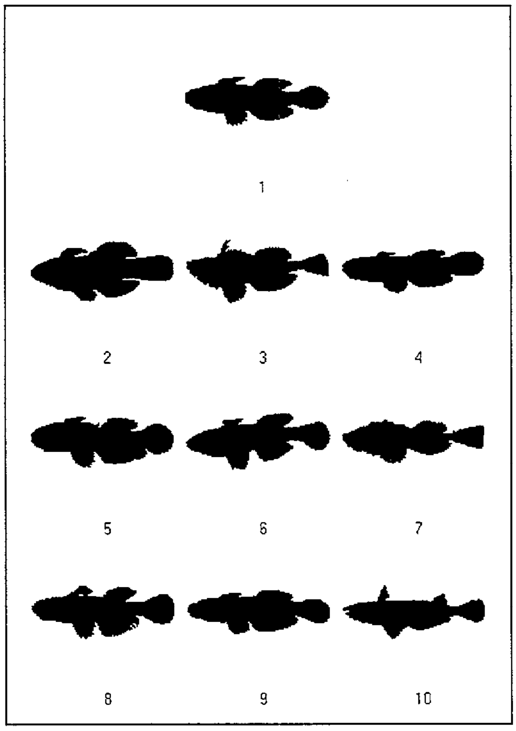


Figure 5.38: Example of the results of searching for similar shapes for shape 635.

order	shape number	weighted energy
1	635	0.00
2	969	3470.31
3	661	3674.35
4	977	3887.81
5	640	3966.25
6	645	4142.67
7	659	4195.95
8	638	4236.88
9	949	4599.64
10	836	4732.11

Table 5.7: Lowest weighted energy values for shape 635.

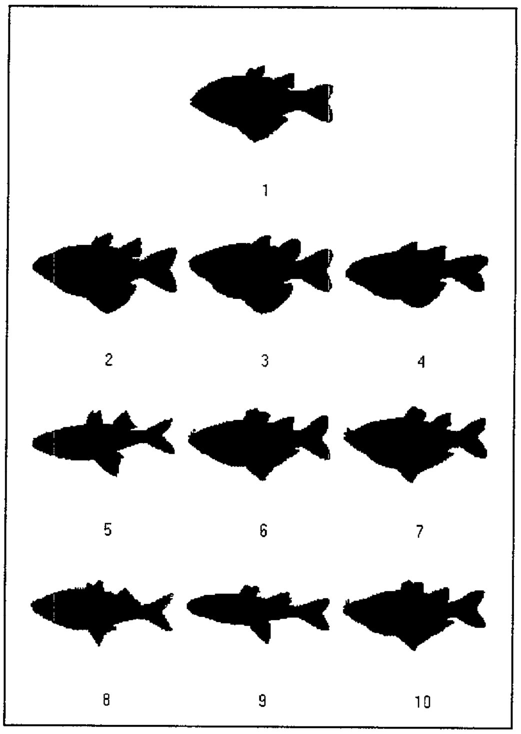


Figure 5.39: Example of the results of searching for similar shapes for shape 554.

order	shape number	weighted energy
1	554	0.00
2	553	1279.90
3	552	1535.54
4	569	1997.10
5	521	2424.76
6	557	2613.10
7	560	2617.37
8	487	2625.74
9	529	2795.74
10	558	2855.33

Table 5.8: Lowest weighted energy values for shape 554.

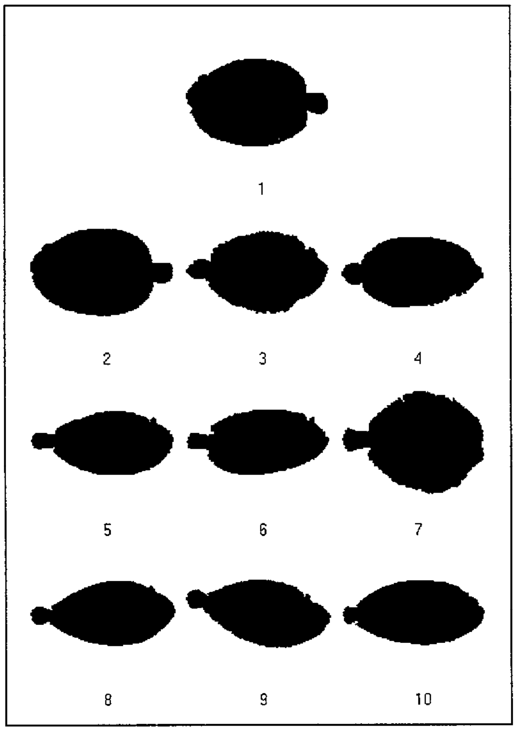


Figure 5.40: Example of the results of searching for similar shapes for shape 634.

order	shape number	weighted energy
1	634	0.00
2	646	280.38
3	277	1103.48
4	275	1358.51
5	984	1449.88
6	938	1517.06
7	276	1710.50
8	696	1750.53
9	695	1797.67
10	988	1897.56

Table 5.9: Lowest weighted energy values for shape 634.

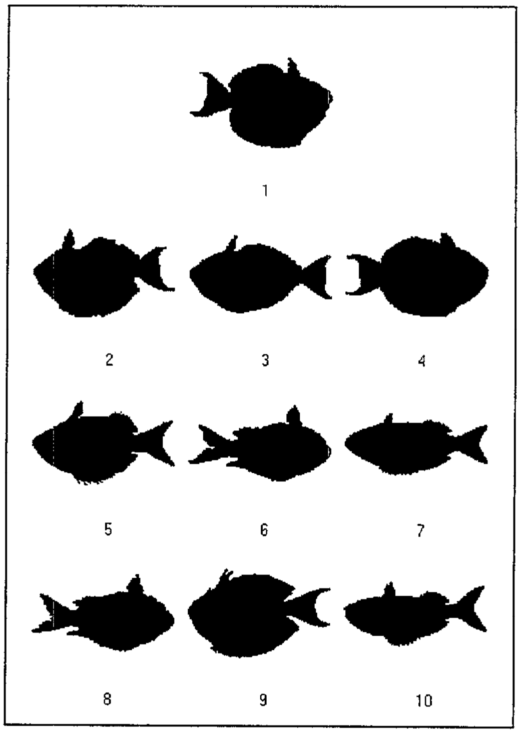


Figure 5.41: Example of the results of searching for similar shapes for shape 378.

order	shape number	weighted energy
1	378	0.00
2	372	1818.41
3	373	2213.36
4	381	2438.41
5	863	2619.30
6	244	2839.06
7	851	3332.50
8	222	3394.02
9	371	3509.69
10	856	3602.97

Table 5.10: Lowest weighted energy values for shape 378.

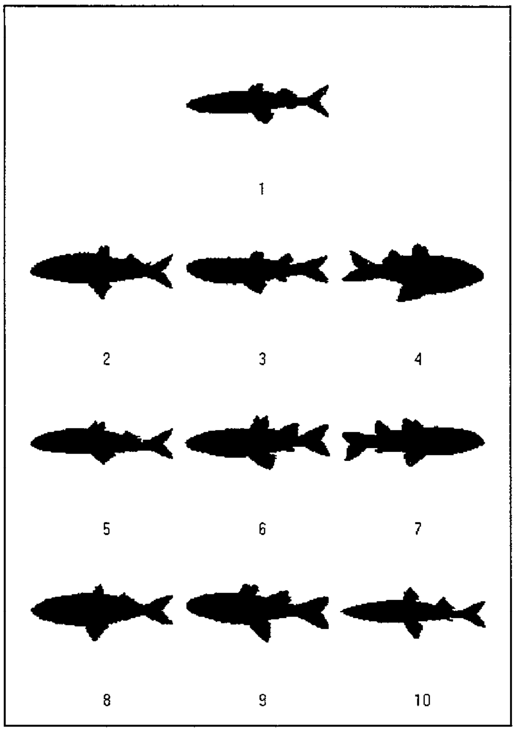


Figure 5.42: Example of the results of searching for similar shapes for shape 514.

order	shape number	weighted energy
1	514	0.00
2	485	1336.25
3	505	1427.80
4	532	1899.20
5	481	1978.40
6	500	2079.59
7	498	2174.04
8	761	2272.73
9	529	2358.54
10	511	2484.58

Table 5.11: Lowest weighted energy values for shape 514.

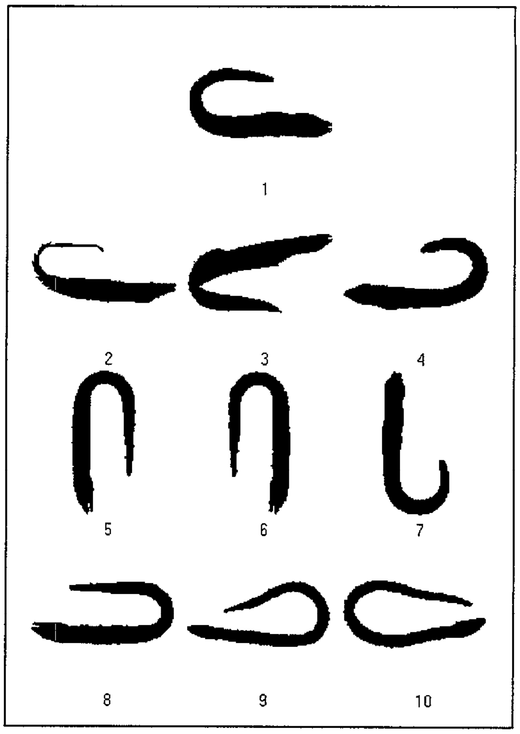


Figure 5.43: Example of the results of searching for similar shapes for shape 11.

order	shape number	weighted energy
1	11	0.00
2	385	1508.93
3	570	1522.45
4	67	1549.49
5	79	1910.27
6	183	2094.56
7	186	2453.26
8	181	2516.63
9	103	4354.88
10	83	7700.93

Table 5.12: Lowest weighted energy values for shape 11.

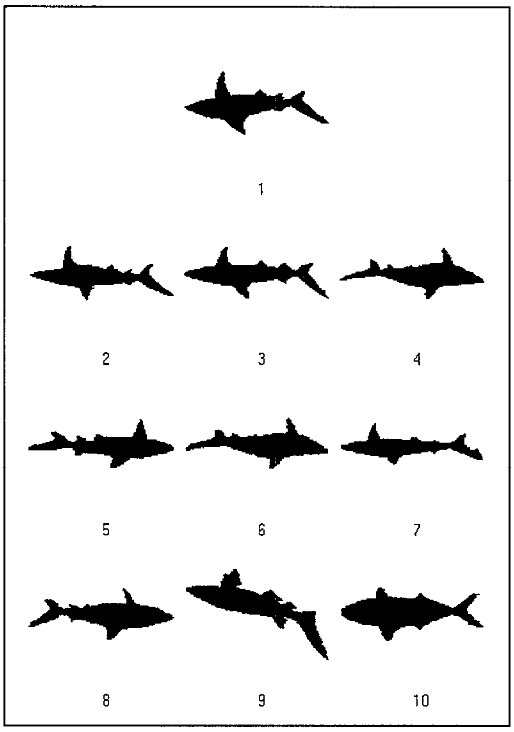


Figure 5.44: Example of the results of searching for similar shapes for shape 716.

order	shape number	weighted energy
1	716	0.00
2	718	1801.08
3	717	1884.89
4	4	2251.81
5	3	2327.86
6	196	2378.62
7	720	3420.78
8	5	5102.87
9	449	6222.79
10	352	6598.08

Table 5.13: Lowest weighted energy values for shape 716.

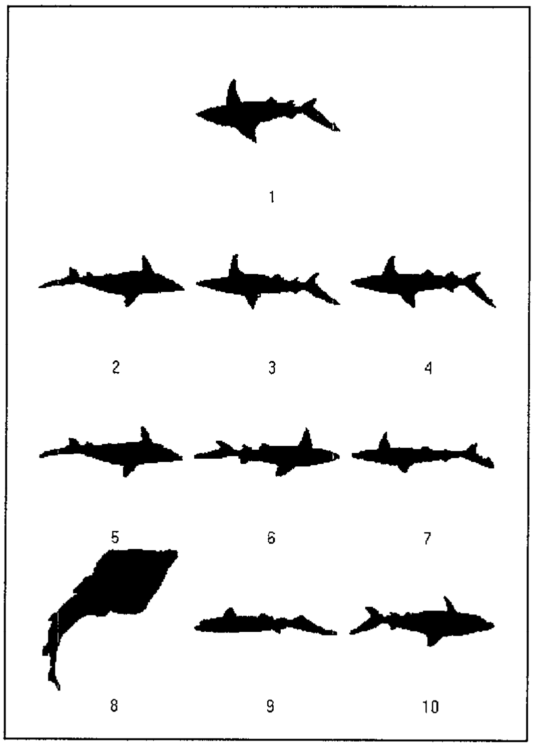


Figure 5.45: Example of the results of searching for similar shapes for shape 716 when the weighting function is not considered.

order	shape number	energy
1	716	0.00
2	4	6615.42
3	718	7060.50
4	717	7287.06
5	196	7885.93
6	3	8166.41
7	720	11090.04
8	55	15679.68
9	704	15803.00
10	5	15851.10

Table 5.14: Lowest energy values for shape 716.

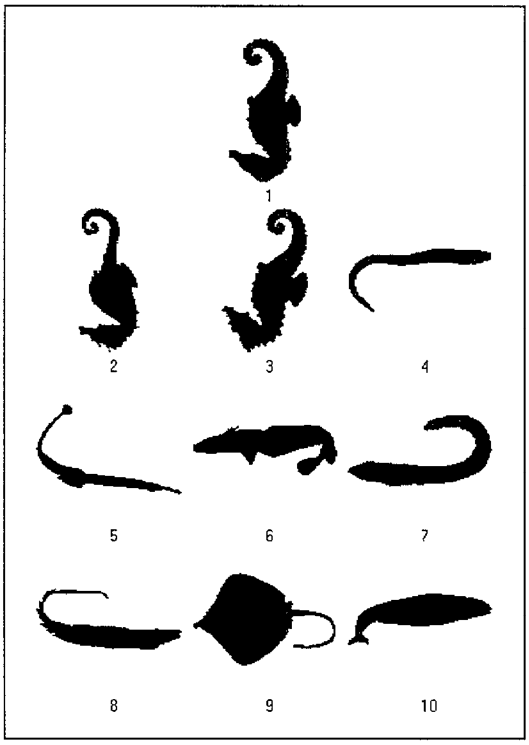


Figure 5.46: Example of the results of searching for similar shapes for shape 779.

order	shape number	weighted energy
1	779	0.00
2	780	3086.65
3	616	5116.34
4	81	12331.39
5	613	15286.41
6	593	15423.97
7	67	16516.39
8	385	17114.49
9	50	18235.94
10	197	18282.40

Table 5.15: Lowest weighted energy values for shape 779.

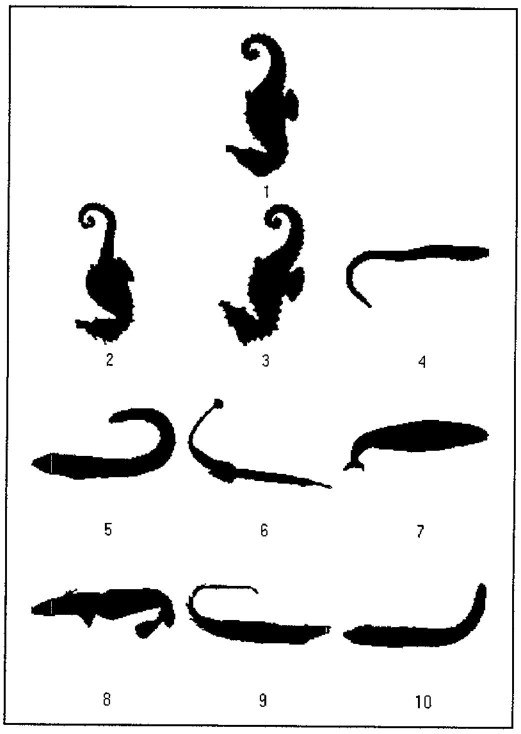


Figure 5.47: Example of the results of searching for similar shapes for shape 779 when the weighting function is not considered.

order	shape number	energy
1	779	0.00
2	780	12165.00
3	616	17853.65
4	81	24285.05
5	67	29042.19
6	613	29723.95
7	197	30290.62
8	593	30792.44
9	385	31127.42
10	80	33290.45

Table 5.16: Lowest energy values for shape 779.

5.6 Characterization of partially occluded shapes

One of the most important problems when trying to identify an object is the fact that it may not be entirely visible in the scene we are working on. When identifying shapes, the object is not always completely available for its recognition, or several overlapping objects may be present in the image. This makes the possibility of occlusion arise in such a way that only some parts of the contour of each object are suitable to describe it. Thus, only some segments of the contour are suitable to be compared with those belonging to a certain database in order to label the object as a member of a certain shape category. In this section, we try to identify objects in the presence of occlusion from a one-dimensional representation of their contours in the form of orientation functions.

Several works can be found in the literature trying to solve this problem with different approaches: general methods [TMV85], a cluster-structure algorithm [BM87], neural networks [KYS96], curvature scale space [Mok97], etc. In this work, the modified Newton filters are used to estimate edge orientation and build a shape representation. If we work with one-dimensional representations of two-dimensional shapes, these contour-based representations must be fitted. When the objects are to be recognized regardless of their size, we cannot determine a priori what proportion of their contours is represented by the visible part of the object. In [AAM01], as well as in [ZR72], Fourier coefficients are used to associate the orientation functions of two shapes. However, this kind of representations describe complete shapes and are not suitable if we wish to relate some sections of each one of the signals. This forces us to search for new approaches, different to those used when complete objects were visible.

5.6.1 Segment extraction

Once we have obtained the orientation functions for two images where common parts are visible, we must determine exactly where these parts are located in both images. We must take into account that they may be continuous sections of the orientation functions or not, thus containing separate segments. Even more, the distance between these segments may be, and usually will, different in both orientation signals, since the uncommon parts between common ones will have different lengths in both signals.

Firstly, we must extract the matching parts in both images, taking into account the Euclidean transformations which may have been produced, i.e. translations and rotations. We assume that the scale is the same for both images. As many small common parts may be found and those straight lines which may appear, e.g. with pseudo-polygonal shapes, may be matched in several parts of the contour, we must select the longest one to fix the rotation which relates both frames. This will be determined by the constant difference between the coupled points in both orientation functions.

After setting the values for the angle, we can proceed looking for other common areas which must fulfill the constraint of the initial rotation angle, since we assume that the object has only undergone a rigid transformation. Since more than one object may be present in the image, the normalization in the length of orientation functions does not assure a direct correspondence between both signals. Thus, it is not useful to adjust the length of the signals to a common value.

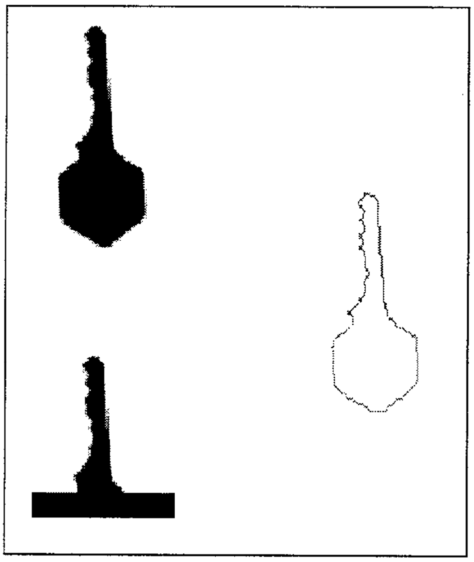


Figure 5.48: Coincident part of two different contours.

A minimum limit must be set in the length of the parts to consider them as significant for shape recognition. In fact, numerous pieces of the orientation function of an occluded object will probably fit in the orientation function of a complete object. Thus, the longest part will determine to which object a segment must be associated. To select such segment, the difference between the values in the orientation functions of both pieces must be less than a certain threshold T_1 along the whole sub-signal. Let $f^r = \{f_k^r\}_{k=0}^{L_r-1}$ and $g^r = \{g_k^r\}_{k=0}^{L_r-1}$ be the r^{th} coincident segments of f_n and g_n, whose length is L_r:

$$d_i(r) = abs\left((f_i^r - g_i^r) - \frac{\sum_{k=0}^{L_r-1}(f_k^r - g_k^r)}{L_r}\right) < T_1 \qquad \forall 0 \leq i < L_r$$

For practical reasons and in order to avoid that noisy images or acquisition processes affect the result invalidating a segment because of a single differing point, a certain number of mismatching values are allowed. Once this limit is surpassed, the segment is concluded and a different one is searched for.

The values in a segment in both images will be used to extract the energy function of a coincident part to measure the similarity as follows:

$$E_r(a) = \sum_{k=1}^{\frac{L_r}{2}} \left(f_k^r - g_k^r e^{-i\frac{2\pi ka}{L}}\right)\left(f_k^r - g_k^r e^{-i\frac{2\pi ka}{L}}\right)^* \tag{5.19}$$

And if we use the weighting function $w(.)$, the energy function in equation (5.19) is transformed into the following weighted energy function:

$$E_r'(a) = \sum_{k=1}^{\frac{L_r}{2}} w\left(\frac{2k}{L_r}\right)\left(f_k^r - g_k^r e^{-i\frac{2\pi ka}{L}}\right)\left(f_k^r - g_k^r e^{-i\frac{2\pi ka}{L}}\right)^*$$

5.6.2 Segment association

If we want to extract other parts which also correspond to the same object, but which are not contiguous to the previous ones, two parameters must be considered. Firstly, the difference in the orientation must be approximately the same for all couples of segments, since we assume that the objects are rigid. As long as the difference is lower than the threshold T_2, the parts to be compared are considered as coincident, as shown in figure 5.49. In order to compare the r^{th} and the s^{th} couples of segments, once each segment has been identified as coincident with its corresponding couple, we proceed as follows:

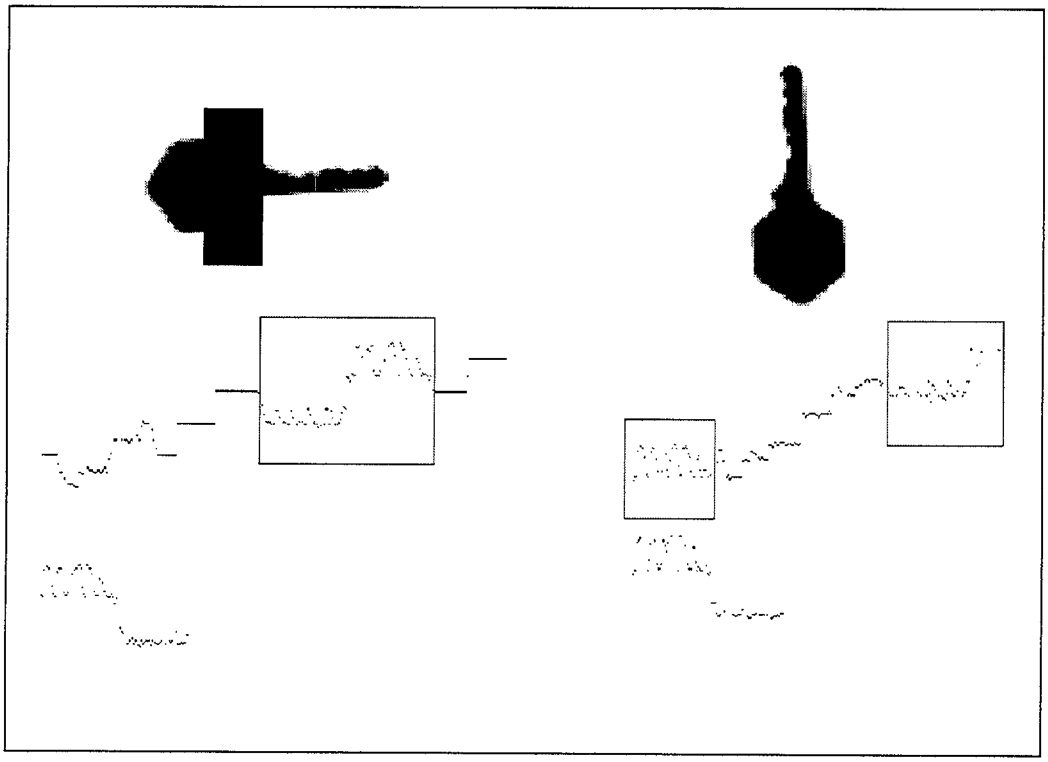


Figure 5.49: Coincident part in the orientation functions.

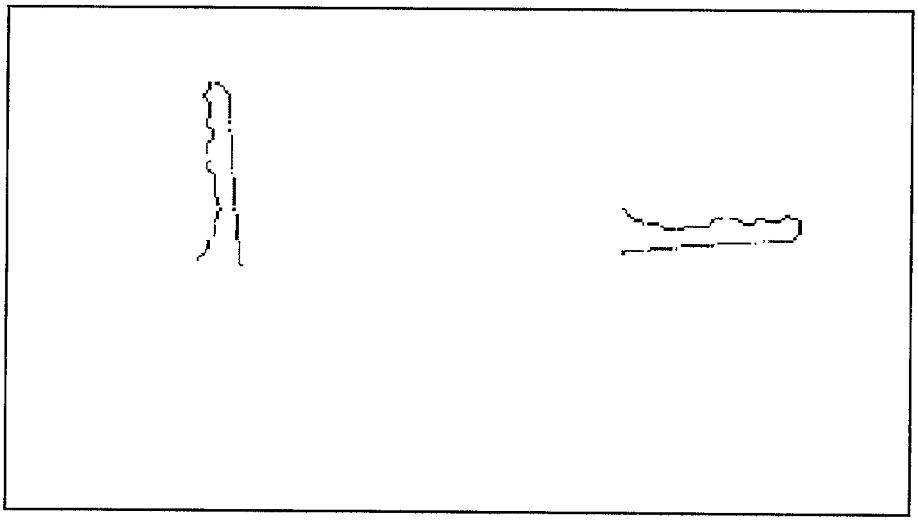


Figure 5.50: Segment identification.

Firstly, we couple the segments of the orientation functions f_n and g_n, f^r and g^r respectively, which satisfy the restrictions in the previous section and whose length L_r must be the same:

$$f^r = \{f_k^r\}_{k=0}^{L_r-1} \quad \leftrightarrow \quad g^r = \{g_k^r\}_{k=0}^{L_r-1}$$

Secondly, we calculate the rotation which brings the second one onto the first one. This can be estimated by the difference in the mean values of the orientation functions corresponding to both segments, since the constant $R(f^r, g^r)$ which must be added to the second one to reach the first one is the angle of the rotation:

$$R(f^r, g^r) = \frac{\sum_{k=0}^{L_r-1} (f_k^r - g_k^r)}{L_r}$$

Finally, we compare the rotations for different couples of segments. In case two angles are similar enough, we can still suppose they belong to the same object. Otherwise, they do not:

$$d_o(r,s) = abs\left(R(f^r, g^r) - R(f^s, g^s)\right) < T_2$$

Secondly, the translation from the rotated points of one of the contours to those in the other one must also be the same. Let $\left(x_i^{f^r}, y_i^{f^r}\right)$ be the coordinates of the points of the r^{th} segment in function f_n, the difference between the translations of the points in any two segments (r and s below) must be small enough, more precisely, lower than a given threshold T_3:

Firstly, we calculate the center of mass for every segment f^r, $m(f^r)$ as:

$$m(f^r) = \frac{\sum_{i=0}^{L_r-1} \left(x_i^{f^r}, y_i^{f^r}\right)}{L_r}$$

Secondly, we rotate the second segment of every couple with respect to the same point, and calculate the translation which fits both segments. We rotate all the segments of a given object with respect to the same point in order to be able to compare the translations properly. Otherwise, the shape would not be preserved:

$$m(g) = \frac{\sum_{i=0}^{L-1} (x_i^g, y_i^g)}{L}$$

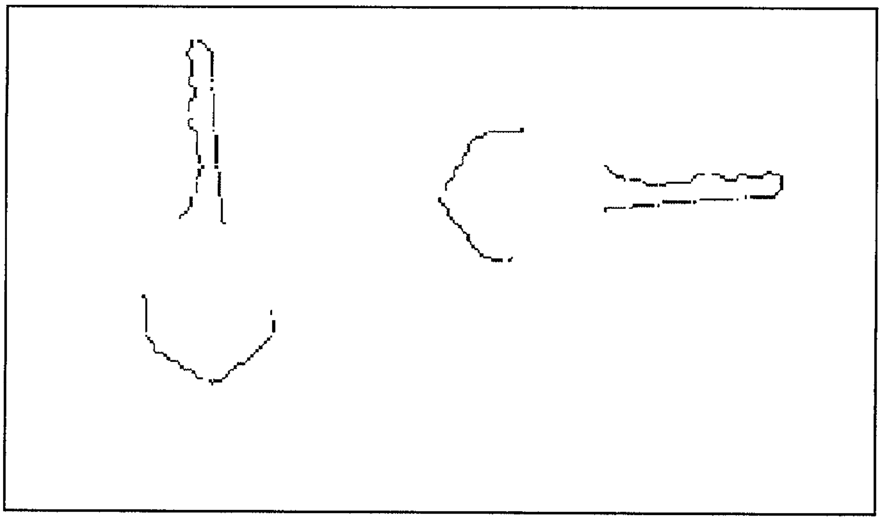


Figure 5.51: Segment association.

The necessary rotation is given by $R(f^r, g^r)$:

$$T(f^r, g^r) = m(f^r) - R(f^r, g^r)m(g^r)$$

And finally, we compare the translations for different couples:

$$d_t(r, s) = ||T(f^r, g^r) - T(f^s, g^s)|| < T_3$$

When both constraints are satisfied, the segments are supposed to belong to the same contour, and thus, they can be considered as visible parts of the same object.

Chapter 6

MOTION ANALYSIS

If we try to analyze the motion of an object, several images must be compared. Global parameters can be extracted from the comparison of the object in different time instants. In fact, the position of the center gives information about the translation, the area covered by the object determines the changes in the size, and the shape of the object will help us decide the rotation angle. In order to process a sequence of frames, we must first study how to compare two of them in isolation.

The study carried out on the contours of a sequence of images from modified Newton filters outputs allows extracting some general parameters of an object, such as its position, orientation and size, in order to analyze its temporal evolution. The combination of one-dimensional processing structures allows building two-dimensional tools, and the combination of two-dimensional structures generates temporal descriptions.

Two different schemes are shown in this chapter. In the first one, local information is only added and compared to extract global information. In the second one, the global description and characterization of the object are used to obtain more accurate results from the mechanism used in the previous chapter for shape characterization.

6.1 Local information in motion analysis

The analysis of a moving object from local information is performed in two steps. Firstly, a local process is carried out in order to detect the changes produced in the neighborhood of the points on the contour. Secondly, those local data are combined and compared in order to extract a global description of the temporal evolution of the object. In this section, we present different approaches which use only local information and the addition of these outputs.

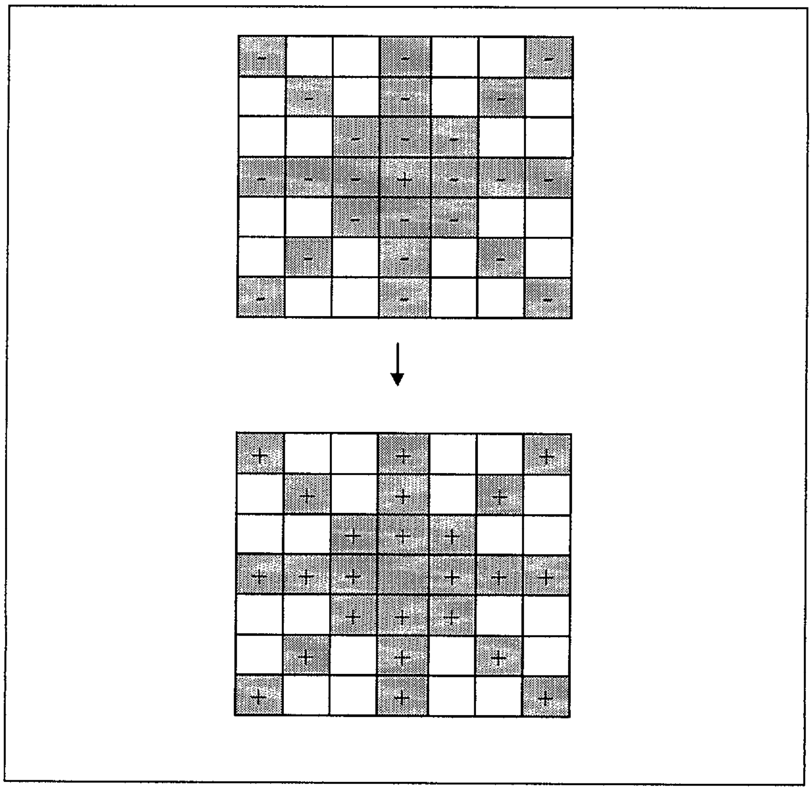
6.1.1 Motion parameter extraction from local information

A previous work by Alemán et al. [ALM97] intended to obtain, from the image of an object located on an artificial retina, an estimation of the center of gravity and size of this object, and from the successive images that arrive to the retina during several temporal samples, an approximation of the direction and speed of its movement in order to be able to follow the object as it moves. The pixels were considered as the quantizing units, as the photoreceptors in the natural system. In this work, the receptive fields were randomly distributed, since the information was processed locally. Each receptive field extracted its own output for direction and speed and the combination of all the outputs gave an idea of what the real trajectory was.

Due to the randomness of the distribution of the receptive fields, whose weights changed as described in figures 6.1 and 6.2, some statistical measures were used for the estimation of size and center of luminance, similarly to the Montecarlo method [Sob74]. For direction and speed, the addition of the outputs determined the global decision.

In [ALM97], as in the current work, eight directions were initially considered, with many receptive fields, i.e. filters, specialized in each of these directions. The detection of edges and their location in the following frame generated an estimation for the direction and speed, as shown in figure 6.3. A temporal integration and a lateral inhibition simulation made the system more robust. The first one avoided drastic changes in the trajectory and the second enhanced the output for the preferred direction, diminishing the probability of erroneous results.

Since the values which were extracted depended on the sample of receptive fields which had been selected, confidence intervals could be used to determine with a certain accuracy the significance of the measures. For example, for a series of measures of the size x_i, the mean $\overline{x}$, standard deviation σ and confidence interval for



-			-			-
	-		-		-	
		-	-	-		
-	-	-	+	-	-	-
		-	-	-		
	-		-		-	
-			-			-

+			+			+
	+		+		+	
		+	+	+		
+	+	+		+	+	+
		+	+	+		
	+		+		+	
+			+			+

Figure 6.1: Weights of the receptive fields before and after the detection of an edge,

significance α are given by equations (6.1), (6.2) and (6.3), respectively:

$$\overline{x} = \frac{1}{N}\sum_{i=0}^{N-1} x_i \tag{6.1}$$

$$\sigma = \sqrt{\frac{\sum_{i=0}^{N-1}(x_i - \overline{x})}{N-1}} \tag{6.2}$$

$$\overline{x} - \sigma t_{\alpha/2} < \mu < \overline{x} + \sigma t_{\alpha/2} \tag{6.3}$$

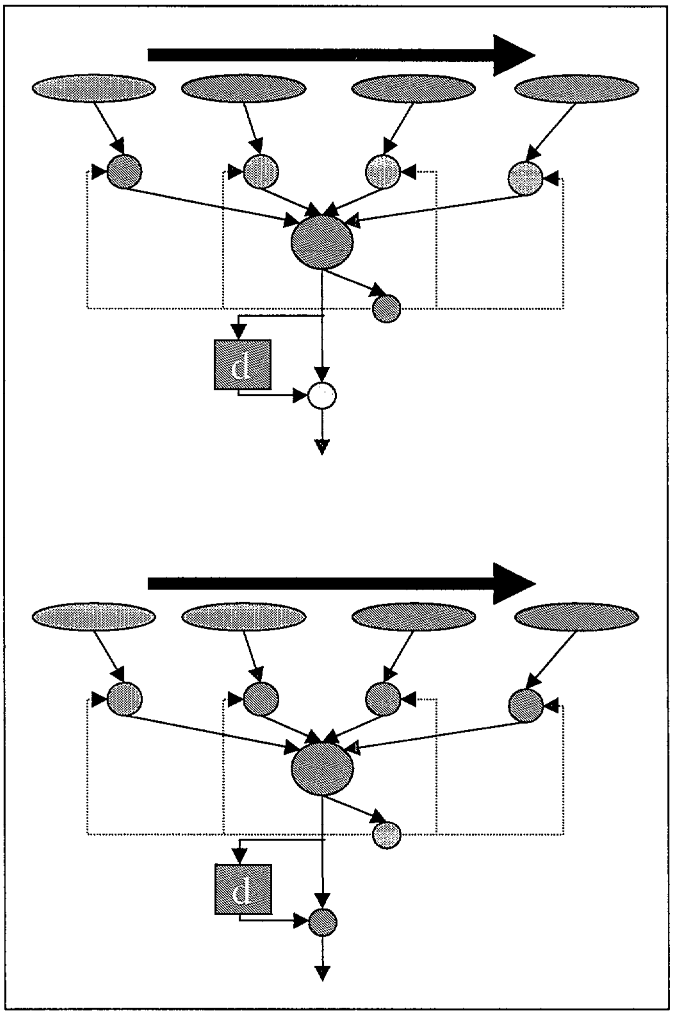


Figure 6.2: Outputs at two instants for a moving edge.

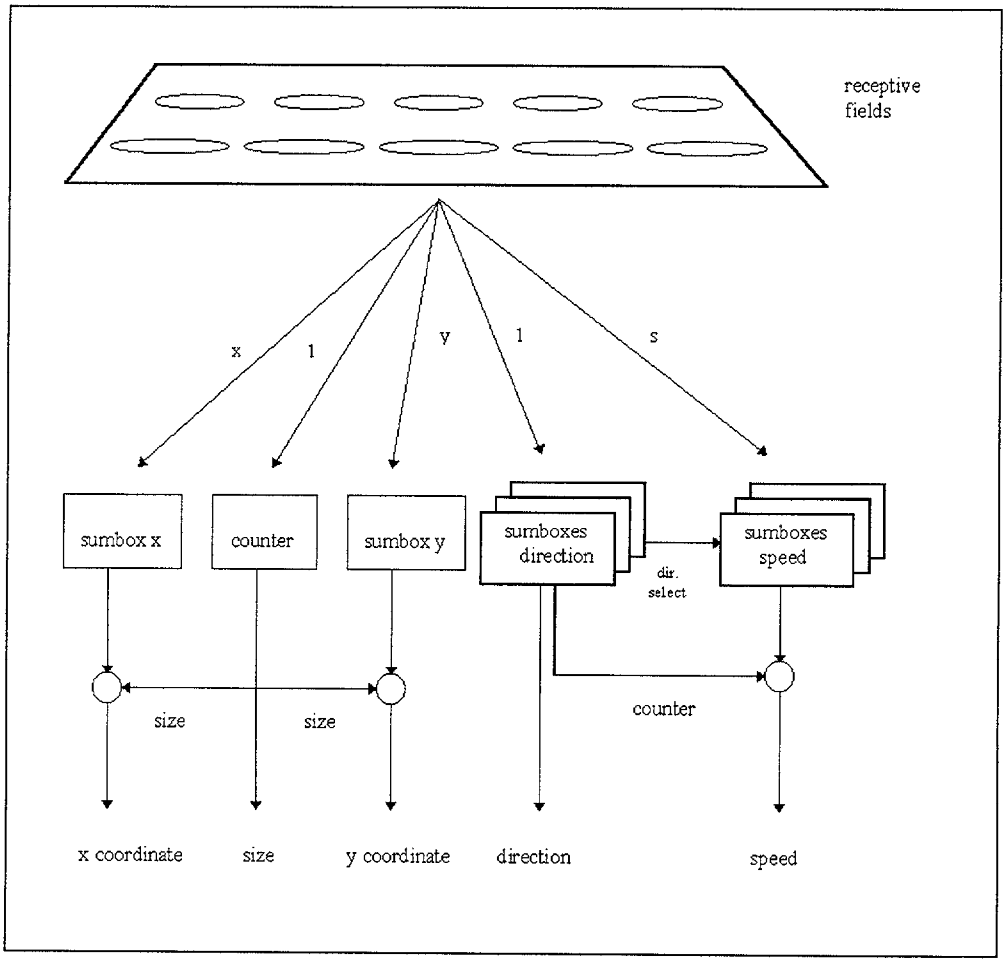


Figure 6.3: Block scheme for motion analysis from local information.

6.1.2 Motion detection using local orientation estimations

The combination of the outputs of the filters specialized in different directions prior to their global analysis allows a much more accurate determination of the orientation of the contours and that enlarges the quantity of orientations to be discriminated. The local comparison of the contours can be improved by considering the more accurate information which can be extracted from the modified Newton filters. In this way, not only eight possibilities are considered, but all those which can be identified with the interpolation process described in the previous chapter. This reduces significantly the ambiguity generated by those borders which can be relocated in many different points in the following image. For each point on the contour, those points in its neighborhood which have the same orientation in the following image are candidates to be its corresponding one, assuming that motion consists only in a rigid translation of the object. When only one candidate is found, the solution is simple, but when multiple choices are available, the ambiguity remains.

Figure 6.4 presents an object which moves in the North-East direction. For every point on the contour in a frame, those points in the neighborhood which present the same orientation are searched for. Using the modified Newton filters and the contour of the object, figure 6.5 shows those points which detect motion in each one of the eight main directions, i.e. those points which have a neighbor in that direction with the same orientation, at a distance lower than 3 pixels for this example. As observed, the most significant output is given by the North-East direction.

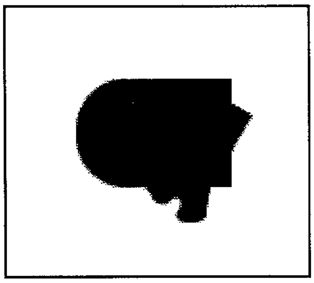


Figure 6.4: Moving object.

However, the study of information globally, even if it is originally processed in a restricted neighborhood for each pixel, provides more accurate information and the possibility of reducing the ambiguities caused by the aperture problem which appears when the receptive fields are much too small for the size of the object.

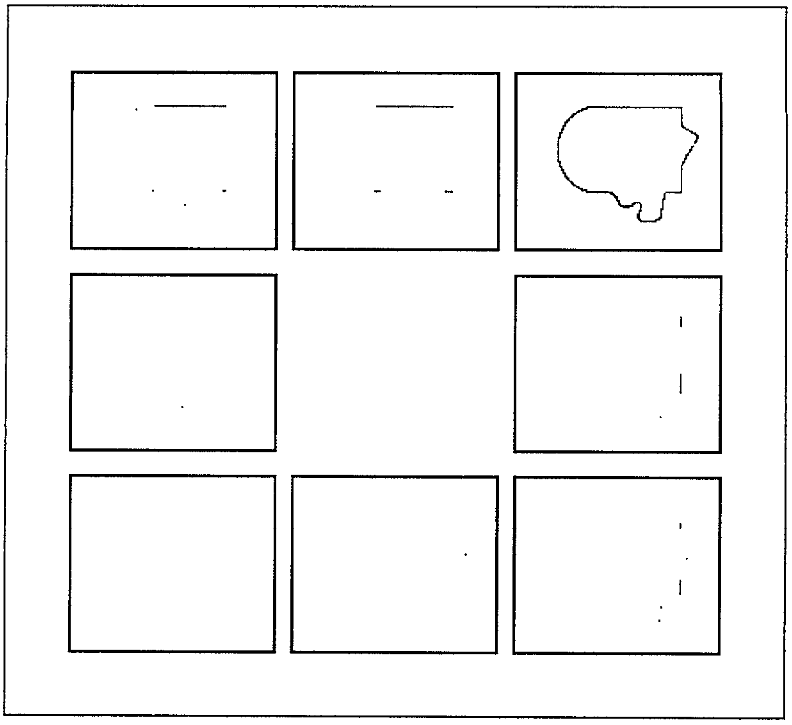


Figure 6.5: Selective motion detection from interpolation of modified Newton filters ouputs.

Furthermore, the previously described mechanism works for translation, but it does not tackle the problems of rotation or scaling. That is the reason why we have used the extraction of the contour as in the previous chapter to study how two different images of the same object can be compared in time.

6.2 Motion analysis from global information

In this section, we start from the results obtained in chapter 5. If we consider a sequence of images corresponding to the acquisition of a moving object as a set of couples of images in which the same object can be identified, the process to relate two contours can be repeatedly used to fit object contours and extract motion parameters.

6.2.1 Object fitting

Once we have identified an object as one of the elements of our database, we may need to bring it to a convenient position. This requires fitting the present object to the reference shape. Firstly, a translation may be required. It can be extracted from the centers of both coupled shapes by subtracting their coordinates. Equations (6.4) and (6.5) determine how we can obtain the center (x_c, y_c) from a function $O(x, y)$ which returns the part of pixel (x, y) which is covered by the object in image I. Let (x_c^1, y_c^1) and (x_c^2, y_c^2) be the centers of both objects, equation (6.6) shows the translation to be performed.

$$x_c = \frac{\sum_{(x,y)\in I} xO(x,y)}{\sum_{(x,y)\in I} O(x,y)} \tag{6.4}$$

$$y_c = \frac{\sum_{(x,y)\in I} yO(x,y)}{\sum_{(x,y)\in I} O(x,y)} \tag{6.5}$$

$$T = (x_c^1 - x_c^2, y_c^1 - y_c^2) \tag{6.6}$$

Secondly, we must consider their sizes. As said before, orientation functions have been normalized to make comparisons. However, we must work with actual sizes, which can be extracted from the segmentation of the images in two regions, object and background. The area covered by the object in the image is given by equation (6.7). The square root of the quotient of both sizes determines the proportional scale we must use, as shown in equation (6.8):

$$A = \sum_{(x,y)\in I} O(x,y) \tag{6.7}$$

$$S = \sqrt{\frac{A^1}{A^2}} \tag{6.8}$$

Finally, the rotation to be performed with respect to the center can be identified from the mean values of both orientation functions. The difference between the mean values, $\overline{f}$ and $\overline{g}$, of two orientation functions corresponding to the same shape allows us to extract the angle θ that should be used to rotate one of them and fit the other one, as shown in equation (6.9).

$$\theta = \overline{f} - \overline{g} \tag{6.9}$$

Nevertheless, a more accurate value for the rotation angle θ can be obtained by minimizing the error function in equation (6.10), where (x_i^1, y_i^1) and (x_i^2, y_i^2) are the coupled points of the contour of both signals whose coordinates are calculated with respect to their respective centers:

$$E_r(\theta) = \sum_{i=1}^{L-1} \left[\left(x_i^1 - \left(x_i^2 \cos\theta + y_i^2 \sin\theta\right)\right)^2 + \left(y_i^1 - \left(-x_i^2 \sin\theta + y_i^2 \cos\theta\right)\right)^2 \right] \tag{6.10}$$

$$\frac{dE_r(\theta)}{d\theta} = 0 \Longrightarrow \sum_{i=1}^{L-1} \left(-x_i^1 x_i^2 \sin\theta + x_i^1 y_i^2 \cos\theta - y_i^1 x_i^2 \cos\theta - y_i^1 y_i^2 \sin\theta\right) = 0$$

$$\tan\theta = \frac{\sum_{i=1}^{L-1} \left(x_i^1 y_i^2 - y_i^1 x_i^2\right)}{\sum_{i=1}^{L-1} \left(x_i^1 x_i^2 + y_i^1 y_i^2\right)}$$

$$\theta = \arctan\left(\frac{\sum_{i=1}^{L-1} \left(x_i^1 y_i^2 - y_i^1 x_i^2\right)}{\sum_{i=1}^{L-1} \left(x_i^1 x_i^2 + y_i^1 y_i^2\right)}\right) \tag{6.11}$$

With these 3 transformations, given by the parameters in equations (6.6), (6.8) and (6.11), we could adapt the position, orientation and distance of an object to fulfill certain conditions, e.g. to be able to grab it. Figures 6.6 and 6.7 show two examples of object fitting with translation, rotation and scaling.

A given point (x, y) in the second object is transformed according to the parameters (T, S, θ) in three steps as follows:

$$(z,t)^1 = (x,y) + T$$

$$(z,t)^2 = \left((z,t)^1 - (x_c^2, y_c^2)\right) S + (x_c^2, y_c^2)$$

$$(z,t)^3 = \begin{pmatrix} \cos\theta & \sin\theta \\ -\sin\theta & \cos\theta \end{pmatrix} \left((z,t)^2 - (x_c^2, y_c^2)\right) + (x_c^2, y_c^2)$$

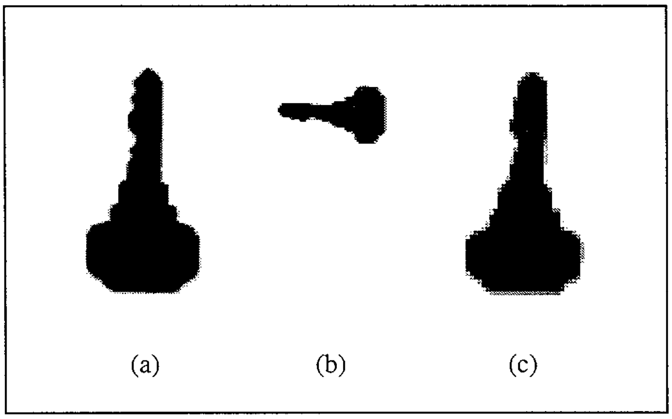


Figure 6.6: (a) and (b) Input images corresponding to the same key. (c) Adaptation of the second image to the first one by means of translation, rotation and scaling.

6.2.2 Motion parameter extraction using global information

The same mechanism which has been used to compare two frames of the same object in different positions and determine the transformation that brings one of them to the other can be used to analyze the temporal evolution of an object in a sequence of consecutive frames. By extracting, for every couple of consecutive frames, the translation of the center of the object, the rotation with respect to the center which the object has undergone and the possible change in the size that may have been produced, it is possible to build a trajectory for the object and represent the changes with regard to the original situation.

In both, object fitting and motion analysis, a certain measure of the correctness of the association must be used. To this effect, we have selected the mean distance from the points in the second contour to those in the first contour, once the transformation has been carried out with the parameters extracted from the frames. This measure is related to the Hausdorff measure, but, instead of selecting the maximum distance, we calculate the average. The mean distance from C_1 to C_2, where C_1 and C_2 are point sets, is given by equation (6.12).

$$d(C_1, C_2) = \frac{\sum_{p_1 \in C_1} \min_{p_2 \in C_2} \|p_1 - p_2\|}{|C_1|} \tag{6.12}$$

As we are working with a sample of points extracted from the contours, we consider the distance from one set to the other, but not vice versa. The lower this measure, the better the matching between a couple of figures. We do not use it as a tool to extract object transformations, but as a test for our parameters. As long as this measure remains low, we can trust in the correctness of the analysis. Other related measures, such as the partial Hausdorff measure, are used for similar purposes by Olson and Huttenlocher [OH97].

Figure 6.9 shows a real sequence of frames and the trajectory which has been extracted. On the left, the initial position of a small toy is shown, as well as the contours for the following frames. On the right, the trajectory is described indicating the changes in the orientation of the object with a vector whose angle corresponds to the rotation of the object and whose length reflects the size changes. Table 6.1 shows the corresponding numerical values for the parameters which have been extracted from this sequence, as well as the mean distance for the object fitting which has been performed. Figure 6.10 and table 6.2 show a second example with a synthetic movie. As observed, the numerical results, whose accuracy is tested with the mean distance, are very satisfactory.

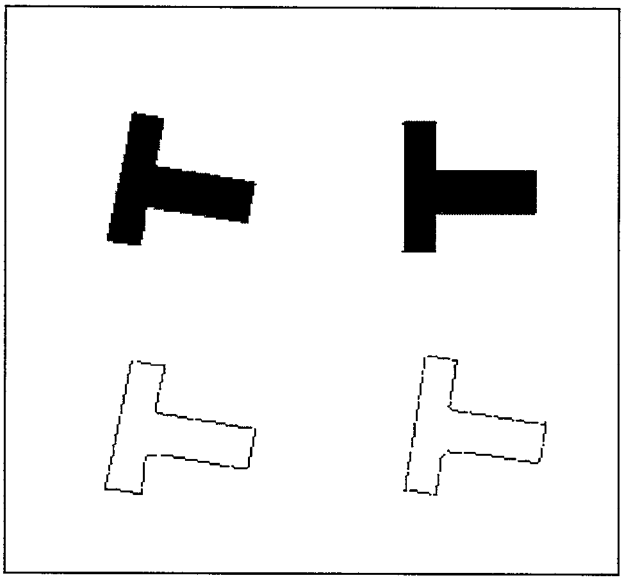


Figure 6.7: Contour fitting of a couple of consecutive images from a motion sequence with translation, rotation and scaling (the contour of the image on the right is adjusted to image on the left).

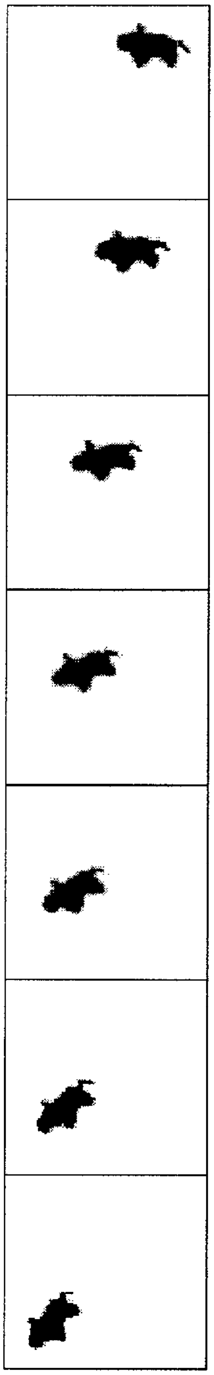


Figure 6.8: Sequence of images of a moving object.

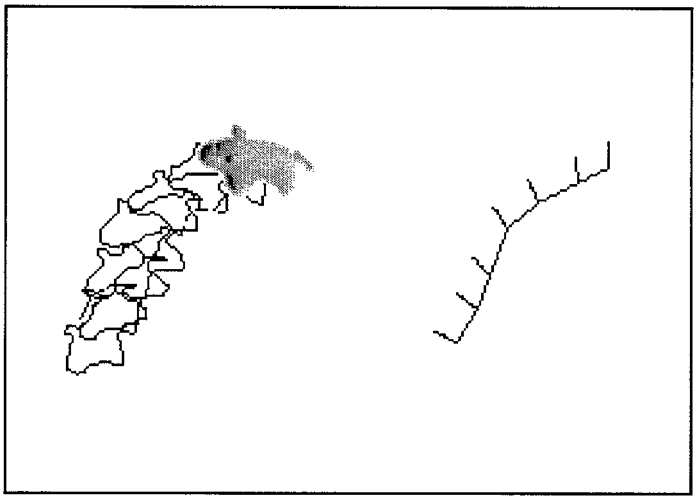


Figure 6.9: Image sequence and temporal evolution. The trajectory of the object is shown on the right and the vectors represented on it indicate the changes in the orientation from the initial situation.

Frames	T_x	T_y	θ	S	d
1-2	-12.6985	-5.5378	0.1332	1.0130	0.4069
2-3	-15.8720	-8.1660	0.2356	0.9602	0.5150
3-4	-12.2985	-10.3341	0.2360	0.9793	0.5317
4-5	-6.7458	-18.3505	0.1306	1.0091	0.5258
5-6	-4.9547	-13.5380	0.2053	0.9880	0.4464
6-7	-7.7208	-12.7441	0.2154	0.9939	0.4866

Table 6.1: Translation (T_x, T_y), rotation angle θ, scaling proportion S and mean distance d for frames in figure 6.9.

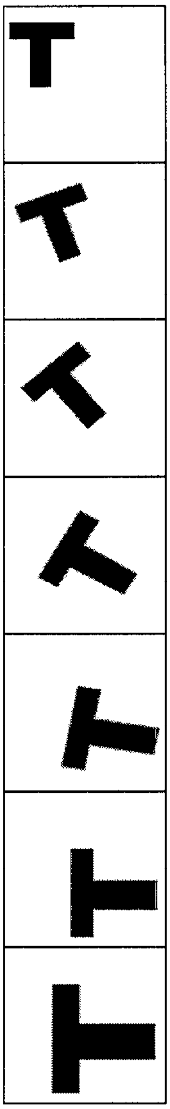


Figure 6.10: Sequence of images of a moving object.

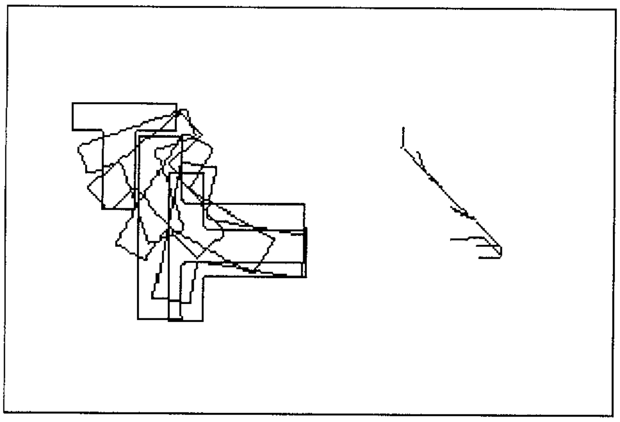


Figure 6.11: Image sequence and temporal evolution. The trajectory of the object is shown on the right and the vectors represented on it indicate the changes in the orientation from the initial situation.

Frames	T_x	T_y	θ	S	d
1-2	11.3974	12.2906	0.3510	1.0445	0.3210
2-3	8.6578	8.3999	0.3733	1.1137	0.4188
3-4	14.7364	13.6461	0.3927	1.2036	0.5265
4-5	12.8263	12.9653	0.3726	1.1242	0.3844
5-6	0.6253	5.0067	0.1721	1.0613	0.8590
6-7	-9.3438	-9.7741	0.0251	1.4544	0.5781

Table 6.2: Translation (T_x, T_y), rotation angle θ, scaling proportion S and mean distance d for frames in figure 6.10.

Chapter 7

TEXTURE CLASSIFICATION

The complete description of an object is not only given by its shape. Thus, the texture in the inner region may be helpful to a large extent when we try to characterize materials, components, agglomerations, etc. Sonka et al. [SHB99] define a texture as something consisting of mutually related elements. Nevertheless, due to its wide variability, it is not simple to give a precise definition. A texture consists of texture primitives or texture elements, sometimes called *texels.* An important problem when dealing with textures is that texture description is scale dependent.

The purpose of this chapter is to present an approach to texture classification based on the description given by the estimation of the orientation in every point of an image. The same kind of Fourier analysis which has been carried out to characterize shapes from their contours can be used to identify the texture which fills a region of an image. Firstly, modified Newton filters are used to extract the orientation of the gradient in every point of the image. Due to the fact that we are not working with the silhouette of a shape, they are applied all over the textured region, without the need of segmenting first object and background.

Since orientation histograms do not univocally characterize a texture, the same representation could have been extracted from different patterns. The combination of a multiscale analysis with the comparison of orientation histograms results in a powerful technique for texture classification which reduces noticeably the ambiguity of considering isolated histograms.

7.1 Orientation histograms

With the modified Newton filters, we are able to estimate accurately the orientation of the gradient at a certain pixel. Similarly, the values for the eight main orientations can be used to estimate the magnitude of this gradient. The interpolation of the outputs of the filters will provide us with a value for the direction of the gradient in every point as well as an estimation of its magnitude. With these estimations, we build a histogram of the orientations by adding the magnitudes of the gradients of the points where the edges present the same orientation. This histogram describes how the orientations are distributed in that region, allowing us to determine the most significant ones, the proportion they represent and their relation in terms of orientation distance and concentration. Figure 7.1 shows an example of a texture and the corresponding orientation histogram which can be used to compare this texture with others. The orientations are grouped into a discrete set of equidistant values with a given cardinal L.

For the orientations, we use the quadratic interpolation polynomial described in chapter 5 for each point (j,k) in the image as shown in equation (7.1), where i is the index of the filter with the highest output f_i and positions $i-1$ and $i+1$ are calculated modulo 8.

$$\begin{aligned} y = & \frac{8(f_{i+1}-f_{i-1})-16(f_i-f_{i-1})}{\pi^2}\left(x-\frac{\pi}{4}(i-1)\right)^2 \\ & + \frac{8(f_i-f_{i-1})-2(f_{i+1}-f_{i-1})}{\pi}\left(x-\frac{\pi}{4}(i-1)\right)+f_{i-1} \end{aligned} \tag{7.1}$$

With this polynomial, the maximum value $o_{j,k}$, which estimates the orientation in that point, is given by:

$$o_{j,k} = round\left(\frac{\left(\frac{4(f_i-f_{i-1})-(f_{i+1}-f_{i-1})}{2[2(f_i-f_{i-1})-(f_{i+1}-f_{i-1})]}\frac{\pi}{4}+\frac{\pi}{4}(i-1)\right)L}{2\pi}\right)$$

It has been rounded to adjust it to a discrete signal. For the magnitude, we use this orientation and substitute it in the polynomial:

$$\begin{aligned} (\nabla I)_{j,k} = & \frac{8(f_{i+1}-f_{i-1})-16(f_i-f_{i-1})}{\pi^2}\left(o_{j,k}-\frac{\pi}{4}(i-1)\right)^2 \\ & + \frac{8(f_i-f_{i-1})-2(f_{i+1}-f_{i-1})}{\pi}\left(o_{j,k}-\frac{\pi}{4}(i-1)\right)+f_{i-1} \end{aligned}$$

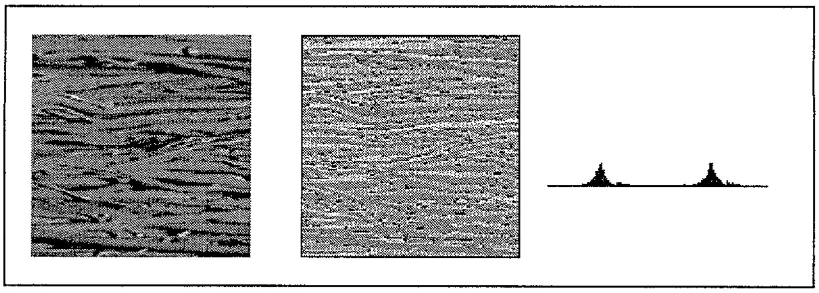


Figure 7.1: Textured region and associated orientation map and orientation histogram.

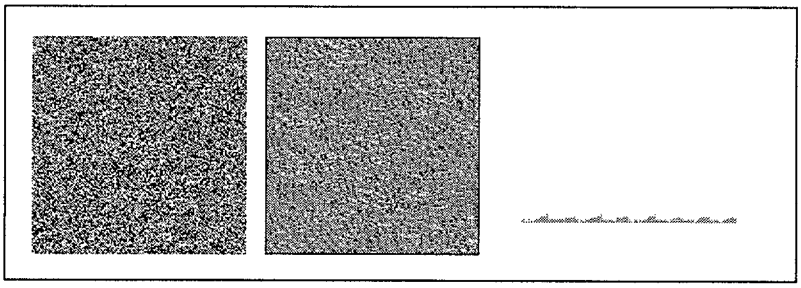


Figure 7.2: Noisy texture with no predominant orientation and corresponding orientation map and orientation histogram.

The values of the histogram h_i are given by the following expression, where $(\nabla I)_{j,k}$ and $o_{j,k}$ are the magnitude and the orientation extracted for point (j,k):

$$h_i = \sum_{\substack{j,k \\ o_{j,k}=i}} (\nabla I)_{j,k}$$

For normalization purposes, the global weight of all positions in the histogram is set to 1, thus dividing each resulting component of the histogram by the sum of all of them:

$$h'_i = \frac{h_i}{\sum\limits_{j=0}^{L-1} h_j}$$

In order to compare two textures, an energy function is built, similar to that used for contour characterization. Nonetheless, in this case, a change in the orientation will only cause a cyclical shift in the histogram and no constant is added to the values. For this reason, the coefficients are modified as follows: let f_n and g_n be the orientation histograms of length L corresponding to the same texture but shifted a position, i.e. the texture has been rotated an angle $\theta = 2\pi a/L$, and let f_k and g_k be the k^{th} Fourier coefficients of these histograms, then the following relationship applies:

$$f_k = g_k e^{-i\frac{2\pi ka}{L}}$$

This makes the energy function simpler than in the case of the orientation functions used for shape characterization in chapter 5, as expressed in the following equation:

$$E(a) = \sum_{k=1}^{\frac{L}{2}} \left(f_k - g_k e^{-i\frac{2\pi ka}{L}}\right)\left(f_k - g_k e^{-i\frac{2\pi ka}{L}}\right)^* \tag{7.2}$$

In addition, the fact that the number of discrete orientations used for the histograms is constant makes the lengths of the signals equal. Consequently, a change in the size of the region where the texture is analyzed will not cause the generation of a different distribution, provided that the histograms are normalized in their global length. As done with other energy functions used for characterization, a weighting function $w(.)$ can be used to emphasize the discrimination, thus obtaining the following expression:

$$E(a) = \sum_{k=1}^{\frac{L}{2}} w\left(\frac{2k}{L}\right)\left(f_k - g_k e^{-i\frac{2\pi ka}{L}}\right)\left(f_k - g_k e^{-i\frac{2\pi ka}{L}}\right)^* \tag{7.3}$$

As mentioned before, the orientation histograms extracted from the textures describe how the different orientations are quantitatively distributed across the region which is studied, but they do not provide any information about the spatial neighborhood of the pixels with a certain orientation. Thus, a completely noisy image, in which all orientations are equally but disorderly present in the image would generate the same histogram as a circle, where the orientation is increased gradually. This forces us to search for a certain technique which complements the information provided by this kind of histograms in order to enhance the recognition capabilities.

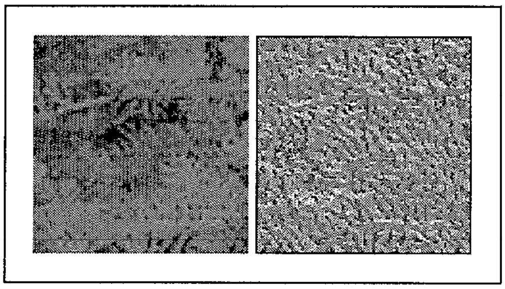


Figure 7.3: Texture and corresponding orientation map.

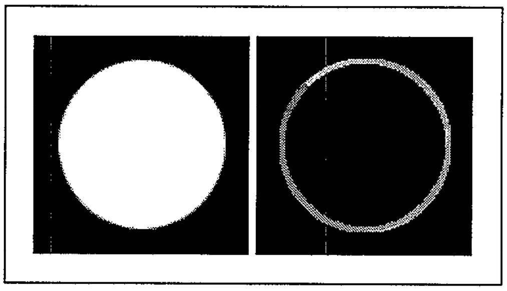


Figure 7.4: Texture and corresponding orientation map.

Figures 7.3 and 7.4 shows the orientation maps for two different textures. The grey-value represents the edge orientation which has been extracted for every point.

A multiscale analysis of the image will provide us with a series of images which represent the evolution of the texture at different scales. In this evolution, the orientations will be differently affected by the others, depending on their spatial proximity. This will allow us to distinguish among textures where orientations are originally distributed in a similar way, but which are actually of different natures.

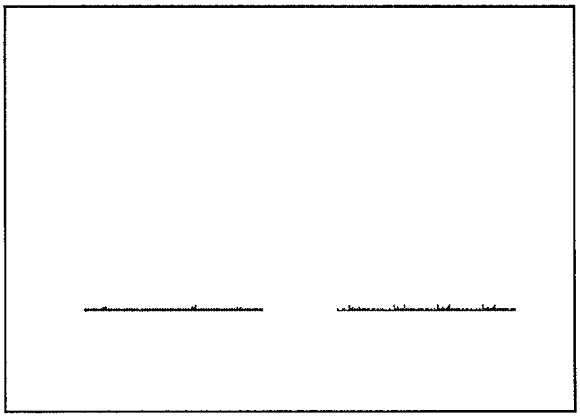


Figure 7.5: Histograms for textures in figures 7.3 and 7.4.

7.2 Multiscale analysis

The interpretation of the information we perceive from the environment depends on the scale we use to process it. For instance, if we look at the sky analyzing the stars independently, we will not be able to realize the relative position of each one of them with respect to the others, avoiding conceiving them as constellations, and not simply as single asters. On the other hand, the interpretation of the information at a very high scale would cause a loss of detailed information about each one of these stars, blurring them into the group.

The information provided by each scale is useful and the study of the same scene at different scales makes it possible to perceive a wider range of realities. Furthermore, elements which are not distinguishable at a certain scale may be clearly distinct at a different one and the rough and detailed information extracted from an image may help us decide when comparing textures.

A multiscale analysis can be determined by a set of transformations $\{T_t\}_{t\geq 0}$, where t represents the scale. Let I be an image, i.e. $I : \Omega \longrightarrow \Re$, where Ω is the domain where the image is defined. In what follows, we will consider $\Omega = \Re^n$ for simplicity in the exposition. $I_t = T_t(I)$ is a new image which corresponds to I at a scale t. For a given image I to which the multiscale analysis is applied, we can extract a histogram $\{h_i^t\}_{i=0,..,L-1}$ which determines the distribution of the orientations of I at the scale t. In this case, the normalization of the values within a histogram is performed with respect to the initial addition, and not with respect to the addition at that scale. In order to compare the histograms of two images, the scale must be first adjusted.

7.2.1 Gaussian multiscale analysis

As said before, a multiscale analysis generates a series of images from an original one, which describe the behavior of the input signal when a certain process is applied to it. Firstly, we will use a Gaussian filter whose evolution at different scales will provide us with a series of values for the proportion of the variation which is preserved when the filter is applied. We use the evolution of the squares of the norms of the gradients:

$$\frac{\int_{\Omega} |\nabla I_t|^2}{\int_{\Omega} |\nabla I_0|^2}$$

With a Gaussian filter, whose properties make it suitable for our purposes as described in [Lin94], we can assure that this value will decrease as we increase the scale. In one dimension, we use a Gaussian kernel is given by:

$$K_t(x) = \frac{1}{\sqrt{4\pi t}} e^{-\frac{x^2}{4t}}$$

and

$$T_t(f)(x) = \int_{\Re} \frac{1}{\sqrt{4\pi t}} e^{-\frac{(x-y)^2}{4t}} f(y)dy$$

where the scale t is related to the standard deviation σ according to the expression:

$$t = \frac{\sigma^2}{2}$$

Afterwards, we quantize it as follows:

$$(K_t)_n = \frac{1}{\sqrt{4\pi t}} e^{-\frac{n^2}{4t}}$$

$$(x * K_t)_m = \sum_{n=-\infty}^{\infty} x_n \frac{1}{\sqrt{4\pi t}} e^{-\frac{(m-n)^2}{4t}}$$

At this point, it is important to consider the relationship between the Gaussian filtering and the heat equation. The heat equation is given by:

$$\frac{\partial u}{\partial t} = \frac{\partial^2 u}{\partial x^2}$$

where $u(t, x)$ is the solution of the equation. Given a signal f, the result of convolving f with the Gaussian filter K_t is equivalent to the solution of the heat equation using f as the initial data:

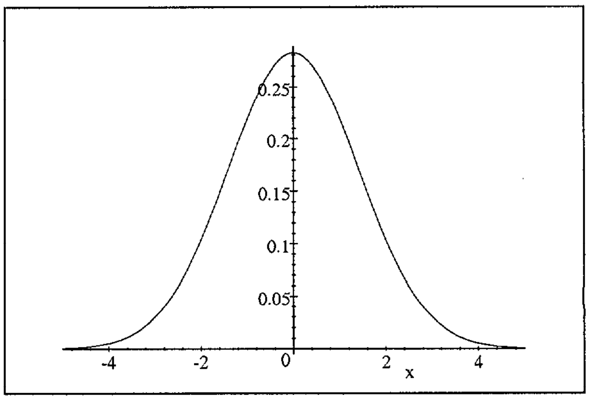


Figure 7.6: Gaussian function for $t = 1$.

$$u(t,s) = K_t * f(x)$$

Considering this relationship, a discrete version of the heat equation can be used to accelerate the approximation of the Gaussian filtering, which results in a recursive scheme in three steps for each direction, as shown below, where I_0 is the original image [AM94]:

$$\begin{array}{ll} I_j^{n+\frac{1}{3}} = I_j^n + vI_{j-1}^{n+\frac{1}{3}} & \forall j \in Z \\ I_j^{n+\frac{2}{3}} = I_j^{n+\frac{1}{3}} + vI_{j+1}^{n+\frac{2}{3}} & \forall j \in Z \\ I_j^{n+1} = \frac{v}{\lambda} I_j^{n+\frac{2}{3}} & \forall j \in Z \end{array} \tag{7.4}$$

This process will be performed by rows and by columns in order to obtain a discrete expression for a two-dimensional Gaussian filtering. Making use of the features of the Gaussian kernels, the result of applying a Gaussian filter with an initial scale t can be used to obtain a Gaussian filtering of the initial image for a different scale without needing to start again from the input. We will discretize the scale considering $\sigma_n = n\sigma_0$ for a given σ_0. Taking into account the relation $\sigma^2 = 2t$, the step size Δt to go from σ_n to σ_{n+1} is given by:

$$\Delta t = \frac{((n+1)\sigma_0)^2}{2} - \frac{(n\sigma_0)^2}{2} = \left(n + \frac{1}{2}\right)(\sigma_0)^2$$

If we use *niter* iterations of the recursive scheme 7.4 to compute I_{σ_n+1} from I_{σ_n}, the discretization scheme for the heat equation is given by:

$$\delta_t^{n+1} = \frac{\left(n+\frac{1}{2}\right)(\sigma_0)^2}{niter}$$

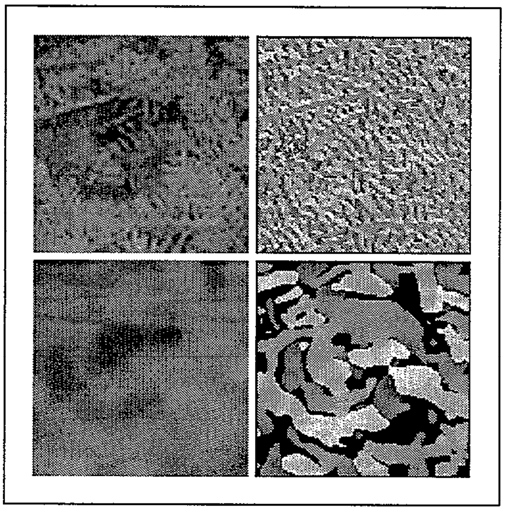


Figure 7.7: Gaussian filtering and change in the orientation map.

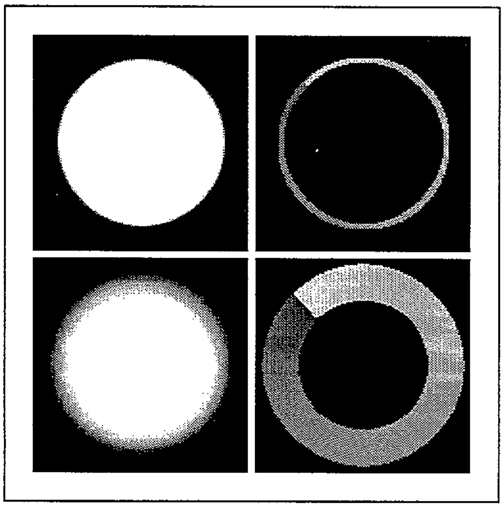


Figure 7.8: Gaussian filtering and change in the orientation map.

7.2.2 Scale estimation

We must take into account that, for a certain texture, the use of different resolutions forces us to apply Gaussian functions with different standard deviations, thus requiring an adaptation stage. Once we have extracted a function of the evolution of the variation at different scales, we can use these factors to compare textures whose histograms are initially very similar. Even if the quantitative distribution of the orientations may be very similar, the spatial distribution will cause a different evolution and interaction, so the factors will differ.

One of the properties of the Gaussian filtering is the relationship between the resolution of two images and the effects of this kind of filters. In fact, the result of applying a Gaussian filter with standard deviation σ to an image with resolution factor x is equivalent to applying a Gaussian filter with standard deviation $k\sigma$ to the same image acquired with a resolution factor kx.

Lemma: Let $I_0(x, y)$, $I'_0(x, y)$ be such that there exists a constant k satisfying that $I'_0(x, y) = I_0(kx, ky)$ $\forall (x, y) \in \Omega$, then

$$I'_t(x, y) = I_{k^2t}(kx, ky)$$

$$\nabla I'_t(x, y) = k \nabla I_{k^2t}(kx, ky)$$

Proof: The result follows from the uniqueness of the solution of the heat equation taking into account that the function $I_{k^2t}(kx, ky)$ is a solution of the heat equation for the initial datum $I'_0(x, y)$.

From this lemma, we deduce that the proportional variation of the squares of the gradient is preserved when the standard deviation of the Gaussian function used for the filter of the scaled image is corrected with the scale factor k, that is:

$$I'(t, x, y) = I(k^2t, kx, ky)$$

$$\frac{\int\limits_{\Omega} \left|\nabla I'_t(x, y)\right|^2 dxdy}{\int\limits_{\Omega} \left|\nabla I'_0(x, y)\right|^2 dxdy} = \frac{k^2 \int\limits_{\Omega} \left|\nabla I_{k^2t}(kx, ky)\right|^2 dxdy}{k^2 \int\limits_{\Omega} \left|\nabla I_0(kx, ky)\right|^2 dxdy} = \frac{\int\limits_{\frac{1}{\Omega}} \left|\nabla I_{k^2t}(x, y)\right|^2 dxdy}{\int\limits_{\frac{1}{\Omega}} \left|\nabla I_0(x, y)\right|^2 dxdy}$$

Let r_n^1 and r_n^2 be the ratios obtained for two textures at the scale $\sigma_n = n\sigma_0$, the best adjusting coefficient k to fit the series of r_n^2 to that of r_n^1 , both consisting of N terms, is given by minimizing the error:

$$E(k) = \frac{1}{N}\sum_{i=0}^{N-1}\left(r_i^1 - kr_i^2\right)^2 \tag{7.5}$$

$$\frac{dE(k)}{dk} = 0 \Longrightarrow \sum_{i=0}^{N-1}\left(r_i^1 - kr_i^2\right) r_i^2 = 0$$

$$k = \frac{\sum_{i=0}^{N-1}\left(r_i^1 r_i^2\right)}{\sum_{i=0}^{N-1}\left(r_i^2\right)^2} \tag{7.6}$$

Once it has been obtained, the values in the second series must be interpolated to compare them with those in the first series.

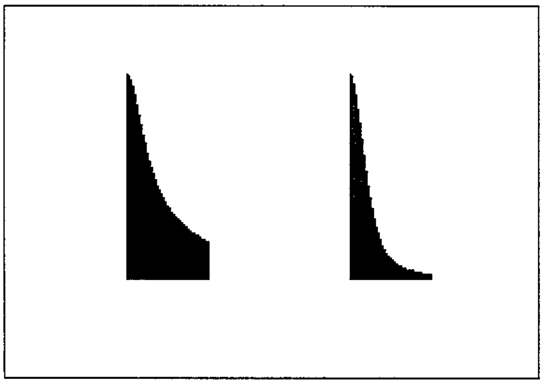


Figure 7.9: Variation of the squares of the gradient for textures in figures 7.7 and 7.8.

The mean error $E(k)$, given by equation (7.5), calculated for the value of k obtained for the best matching scale will determine whether both textures could be related as belonging to the same pattern, but with scales σ and $k\sigma$. If $E'(k)$ is lower than a certain threshold θ, they are assumed to be similar enough and the histogram of the second one can be compared with that of the first after applying Gaussian filters with standard deviations σ and $k\sigma$.

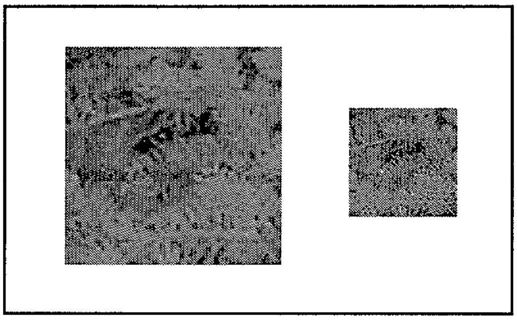


Figure 7.10: Images of the same texture at different resolutions.

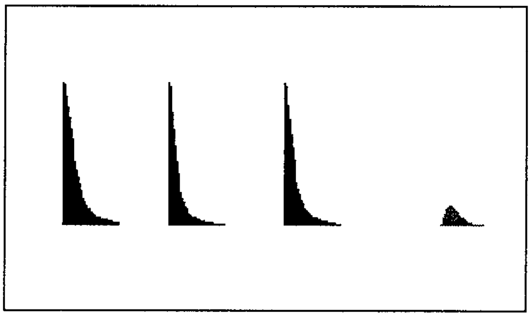


Figure 7.11: Texture scale adjustment and error for textures in image 7.10.

7.3 Texture-based image clustering

This section shows an example of the comparison of a set of textures contained in a database, which is shown in figures 7.12 and 7.13. Using the techniques described in the previous sections, a certain texture is compared with all those in the database and the most similar ones are selected. The similarity between two textures is given by the energy obtained when comparing their orientation histograms. Tables 7.1 and 7.2 show a reference list for texture identification.

We have used a database, made publicly available for research purposes by Columbia and Utrecht Universities, Columbia-Utrecht Reflectance and Texture Database [Col01]. This test set consists of greyscale images and thus, a single histogram is used to represent the orientations of the edges in light intensity. If we were working with color images, a three subchannel mechanism could be used when hue, and not only intensity, is relevant for texture identification.

7.3.1 Clustering with energy values

The following pages show some examples of texture comparison using the technique described in the previous sections. For the image database containing 60 textures of different natures, but visually difficult to classify, one is selected, and the 5 best comparisons are shown. Of course, as the selected image is one of the set, the best match corresponds to itself, and the energy factor is 0.

As done with shape characterization, a weighting function has been used to multiply the factors in the energy function. As in that case, an exponential function makes the values in the lower frequencies more important than those in the higher frequencies, thus giving more importance to the general shape of the histogram than to the small details.

The last example of this series, in figure 7.20, corresponds to a texture which does not belong to the database. It is a white-noise texture generated synthetically and which has been included for this example.

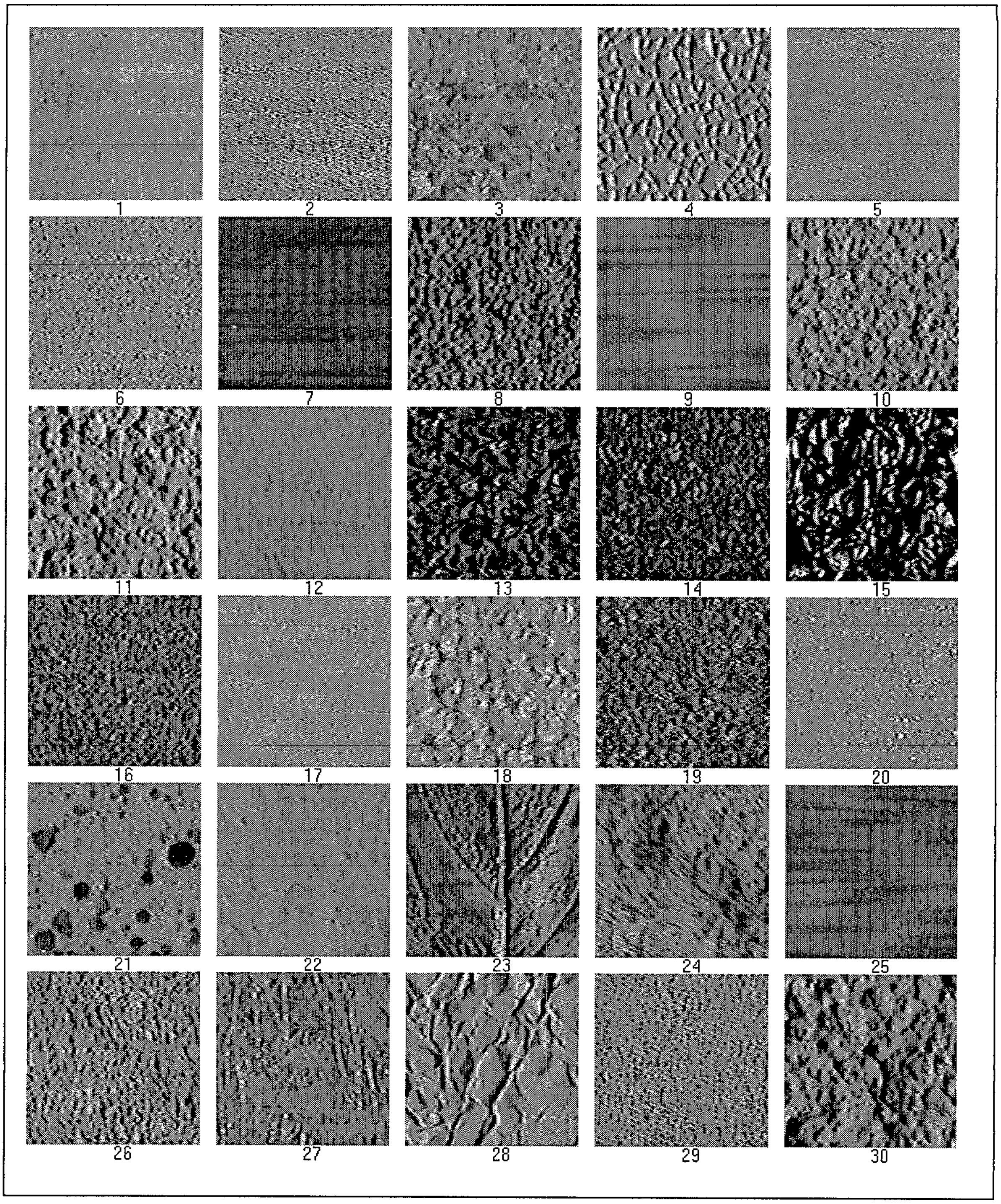


Figure 7.12: Database textures (1).

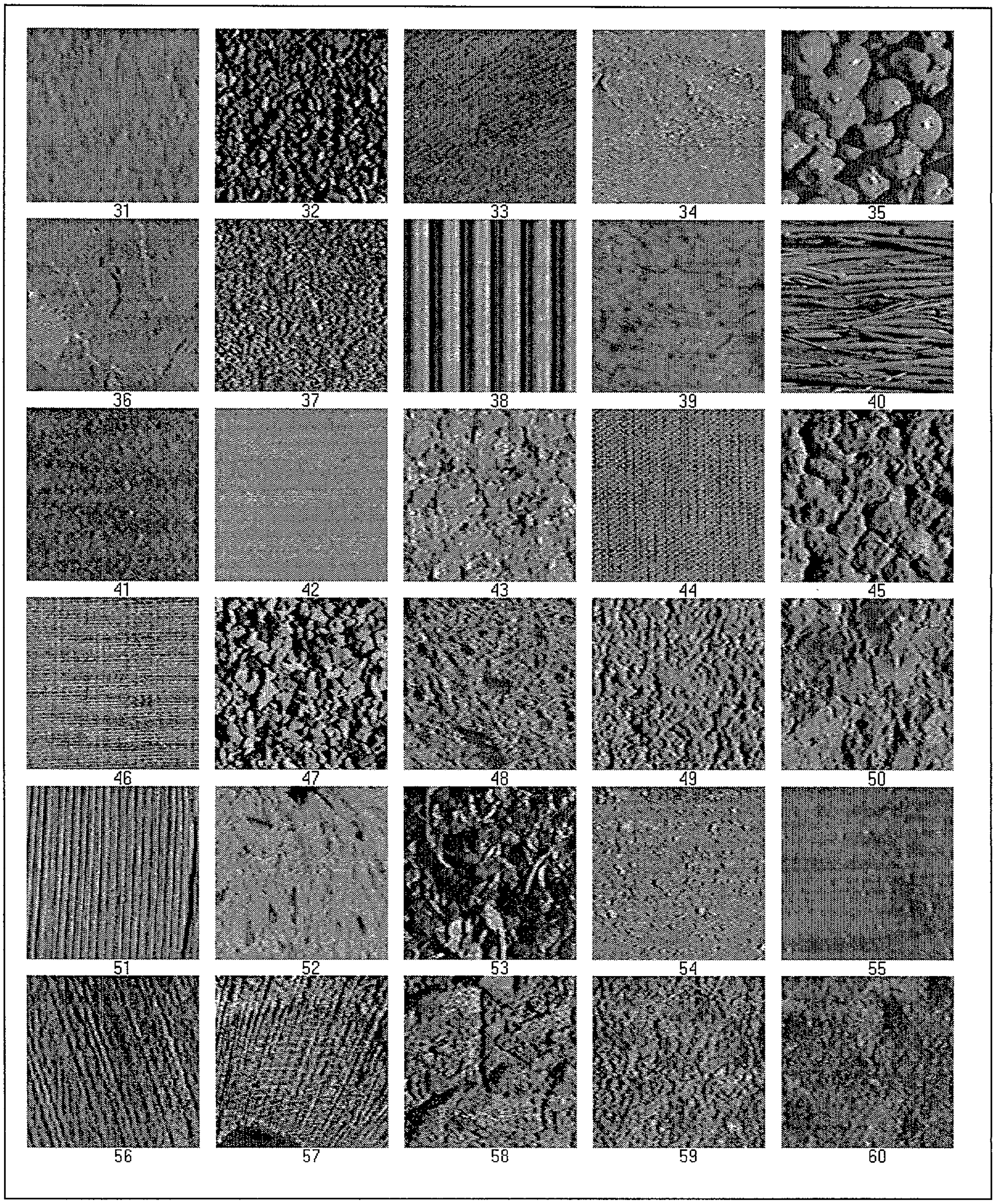


Figure 7.13: Database textures (2).

1	Felt
2	Polyester
3	Terrycloth
4	Rough Plastic
5	Leather
6	Sandpaper
7	Velvet
8	Pebbles
9	Frosted Glas
10	Plaster a
11	Plaster b
12	Rough Paper
13	Artificial Grass
14	Roof Shingle
15	Aluminium Foil
16	Cork
17	Rough Tile
18	Rug a
19	Rug b
20	Styrofoam
21	Sponge
22	Lambswool
23	Lettuce Leaf
24	Rabbit Fur
25	Quarry Tile
26	Loofah
27	Insulation
28	Crumpled Paper
29	Zoomed version of texture 2
30	Zoomed version of texture 11

Table 7.1: Texture reference list (1).

31	Zoomed version of texture 12
32	Zoomed version of texture 14
33	Slate a
34	Slate b
35	Painted Spheres
36	Limestone
37	Brick a
38	Ribbed Paper
39	Human Skin
40	Straw
41	Brick b
42	Corduroy
43	Salt Crystal
44	Linen
45	Concrete a
46	Cotton
47	Stones
48	Brown Bread
49	Concrete b
50	Concrete c
51	Corn Husk
52	White Bread
53	Soleirolia Plant
54	Wood a
55	Orange Peel
56	Wood b
57	Peacock Feather
58	Tree Bark
59	Cracker a
60	Cracker b

Table 7.2: Texture reference list (2).

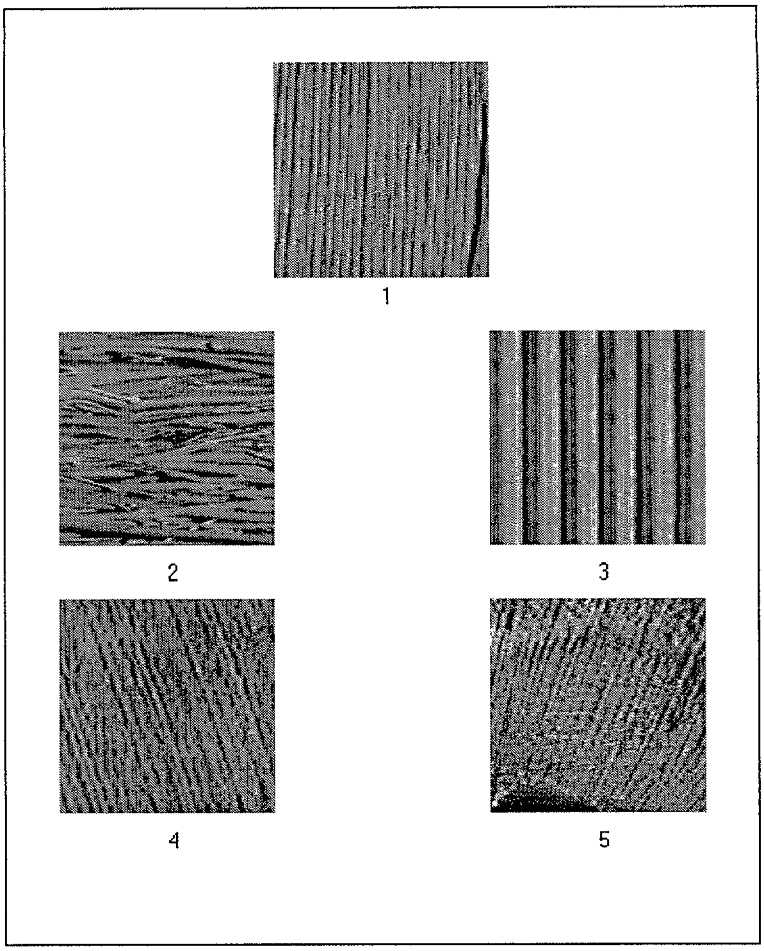


Figure 7.14: Example of the results of searching for similar textures for texture 51 using (7.3).

order	texture number	weighted energy
1	51	0.00
2	40	16.03
3	38	49.21
4	56	118.39
5	57	157.10

Table 7.3: Lowest energy values for texture 51.

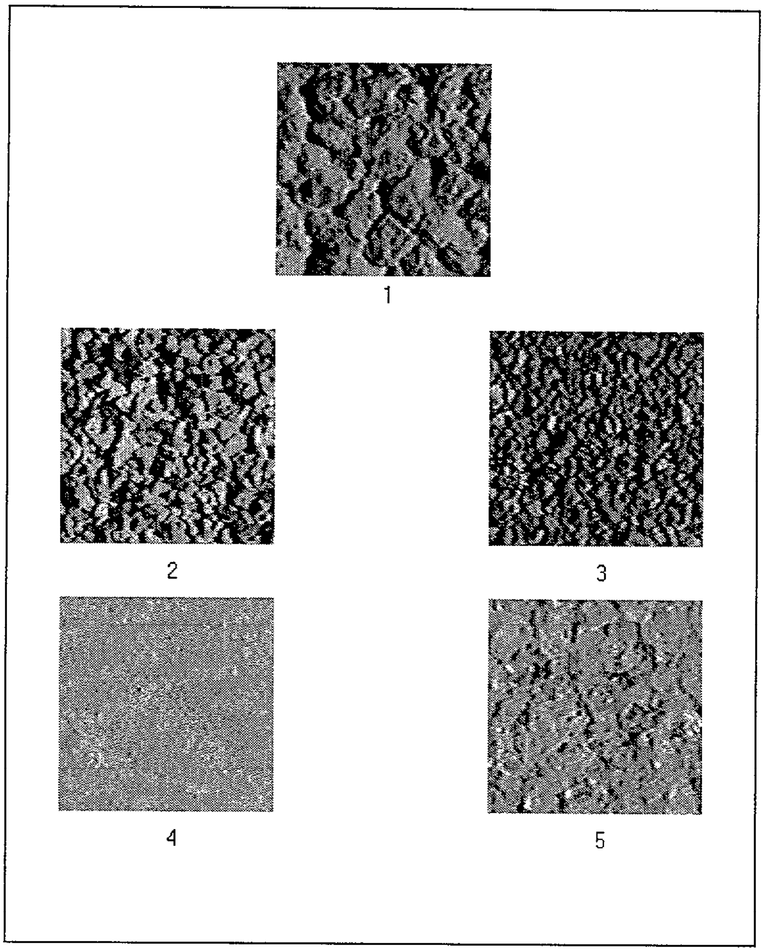


Figure 7.15: Example of the results of searching for similar textures for texture 45 using (7.3).

order	texture number	weighted energy
1	45	0.00
2	47	0.69
3	32	0.91
4	17	0.98
5	43	1.05

Table 7.4: Lowest energy values for texture 45.

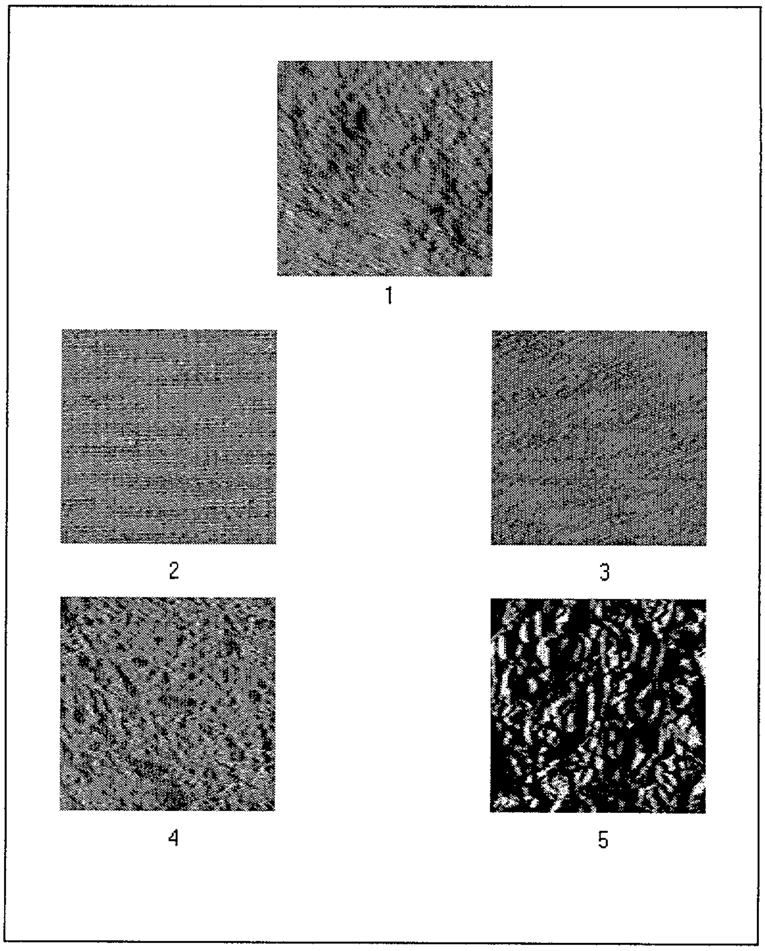


Figure 7.16: Example of the results of searching for similar textures for texture 24 using (7.3).

order	texture number	weighted energy
1	24	0.00
2	46	2.96
3	33	4.53
4	48	6.06
5	15	6.43

Table 7.5: Lowest energy values for texture 24.

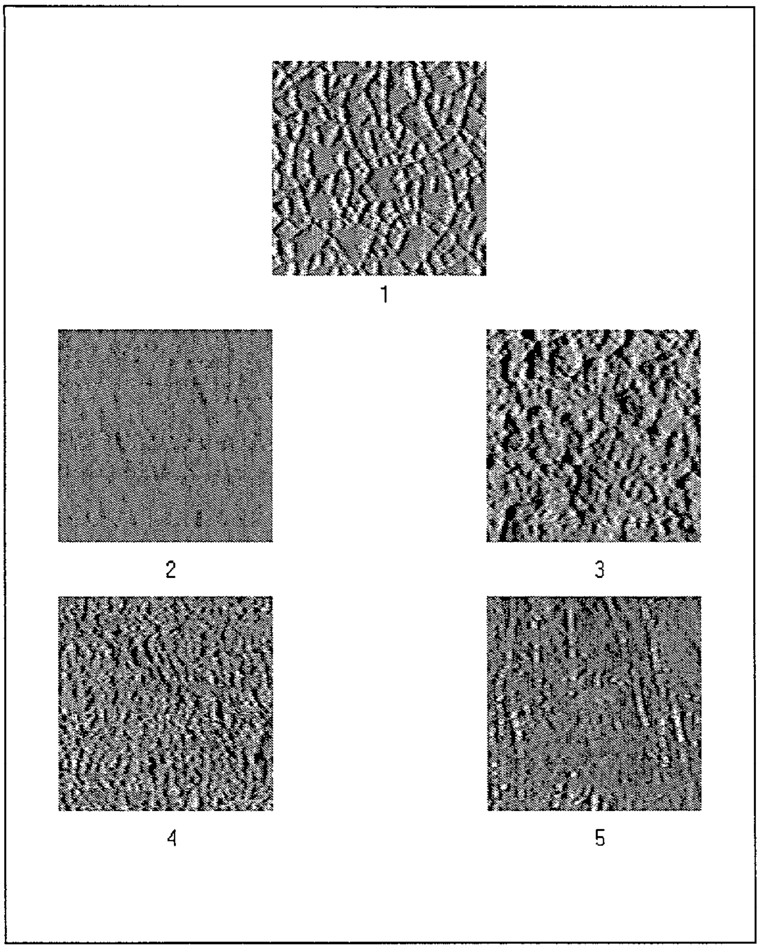


Figure 7.17: Example of the results of searching for similar textures for texture 4 using (7.3).

order	texture number	weighted energy
1	4	0.00
2	31	2.11
3	11	3.77
4	26	4.09
5	27	4.24

Table 7.6: Lowest energy values for texture 4.

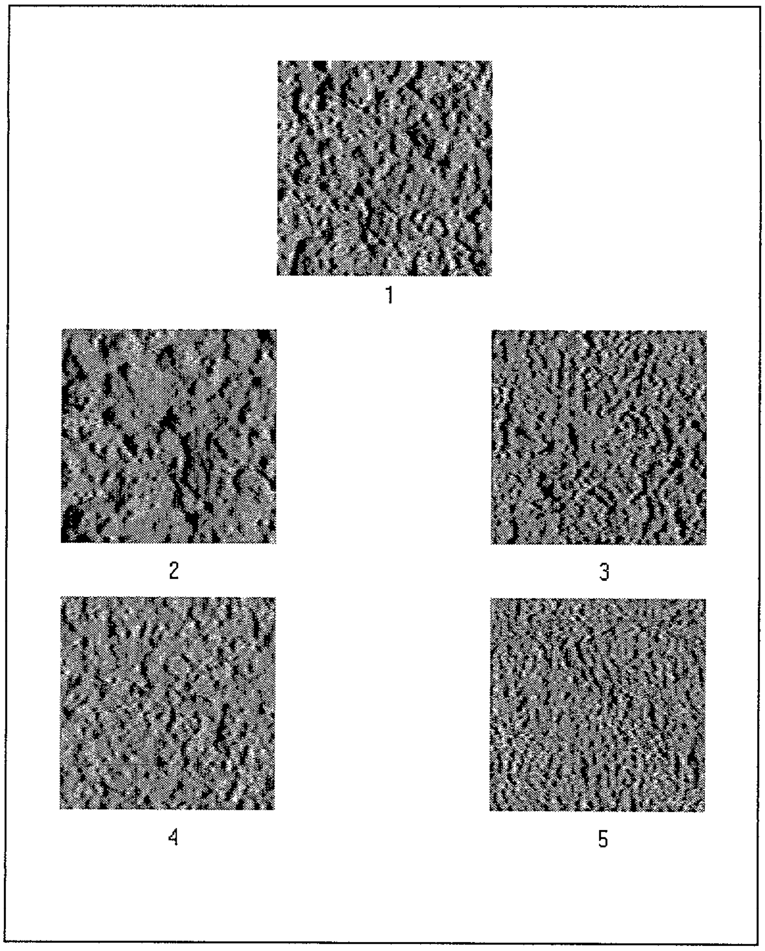


Figure 7.18: Example of the results of searching for similar textures for tetxure 11 using (7.3).

order	texture number	weighted energy
1	11	0.00
2	30	0.59
3	49	0.61
4	10	1.20
5	26	1.62

Table 7.7: Lowest energy values for texture 11.

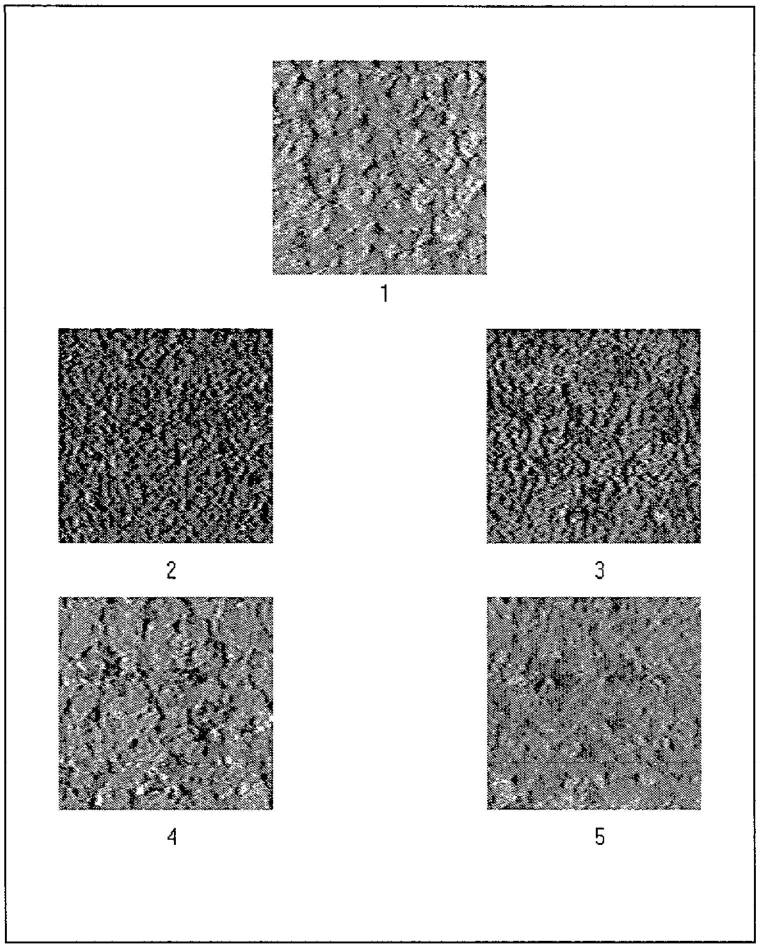


Figure 7.19: Example of the results of searching for similar textures for texture 18 using (7.3).

order	texture number	weighted energy
1	18	0.00
2	16	0.90
3	59	1.15
4	43	1.29
5	3	1.30

Table 7.8: Lowest energy values for texture 18.

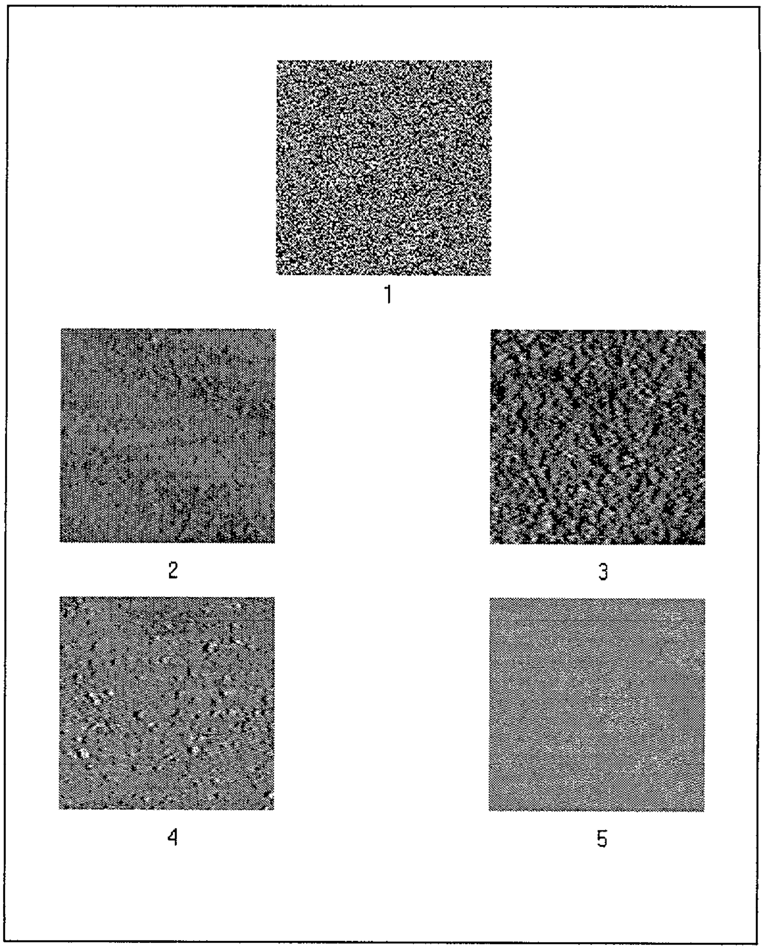


Figure 7.20: Results of searching for similar texture for a white-noise image using (7.3).

order	texture number	weighted energy
1	white noise	0.00
2	41	1.87
3	19	2.06
4	54	2.15
5	42	2.16

Table 7.9: Lowest energy values for a white-noise image.

7.3.2 Two-step clustering: energy values and multiscale analysis

As stated in the previous sections, some textures present similar orientation histograms, even though they are not visually very similar. Although the comparison of the histograms results in low energy values, the evolution of the squares of the gradients allows attaining a more refined discrimination procedure. The following example shows a comparison between two images corresponding to the same texture at different scales. The evolution of the squares of the gradient is similar and it is even more when the matching factor is calculated to adjust one of them to the other. We have calculated the absolute error between both functions to measure quantitatively the dissimilarity, as shown in equation (7.7). As observed, the error resulting is very low when the regions correspond to the same texture at different scales.

$$E(k) = \frac{1}{N}\sum_{i=0}^{N-1} \left| r_i^1 - kr_i^2 \right| \tag{7.7}$$

The previous result has been used to improve the discrimination. Sometimes it is not easy to discriminate the textures when the histograms do not state a clear pattern and can be associated with many textures.

For example, if we compare texture 30 with the others and extract the most similar ones, we obtain a series of results which do not state clear differences between the similar textures and the same texture acquired at a different scale. We apply the multiscale analysis and compare the results for textures 30 and 11 with those for texture 30 and textures 10, 49 and 50. Even if the energy is very similar, the error is higher when the textures are not the same, since the evolution is different as we increase the standard deviation of the Gaussian filtering. This shows how a two-step process (energy calculation and multiscale analysis) can be used to enhance the discrimination when the results in the first step are not clear.

The same process has been carried out with texture 31 (rough paper), whose best approaches are textures 26, 27, 4 and 12. Texture 12 is a zoomed version of texture 31, but the best results are not obtained for it. As observed at the end of this example, textures 4 and 12 (rough plastic and rough paper) are the most similar ones.

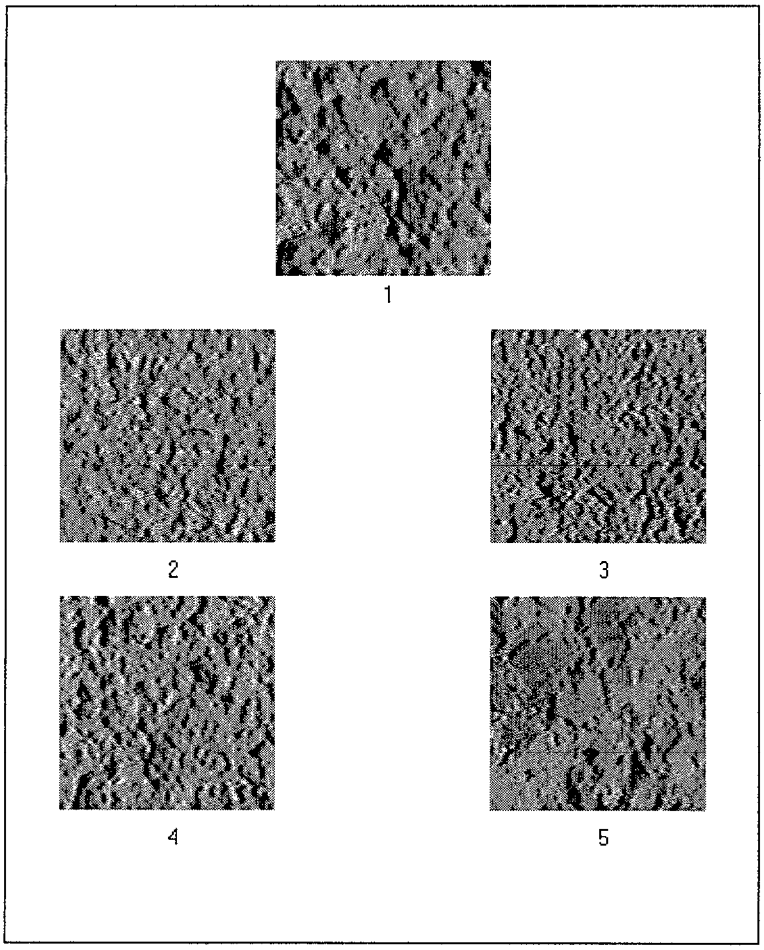


Figure 7.21: Example of the results of searching for similar textures for texture 30 using (7.3).

comparison	energy	weighted energy	quadratic error (*100)	absolute error
30-10	3.50	0.47	3.3053	1.035815
30-49	3.66	0.57	10.2568	1.827342
30-11	4.18	0.59	2.2914	0.877211
30-50	3.85	1.07	3.9379	1.111229

Table 7.10: Comparison of texture 30 with the most similar ones.

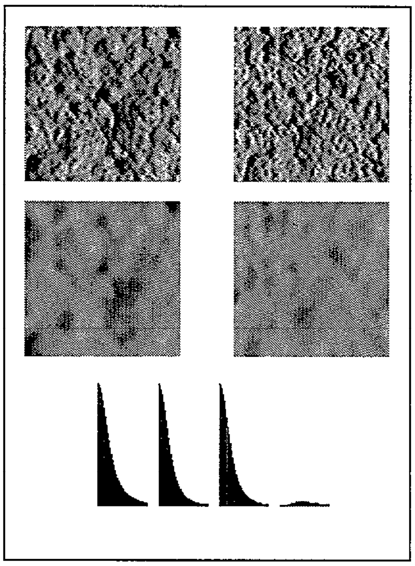


Figure 7.22: Textures 30 and 11.Textures 30 and 11 after Gaussian filtering. Texture scale adjustment using (7.6) and error.

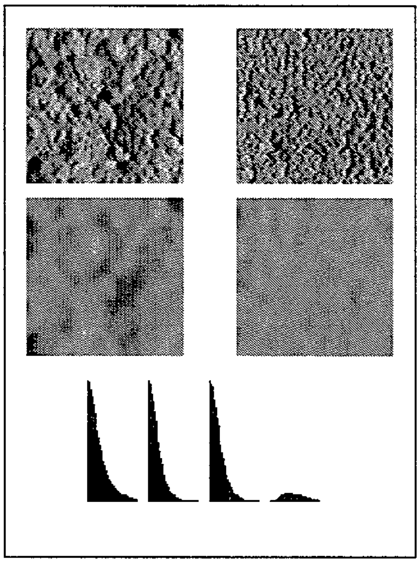


Figure 7.23: Textures 30 and 10.Textures 30 and 10 after Gaussian filtering. Texture scale adjustment using (7.6) and error.

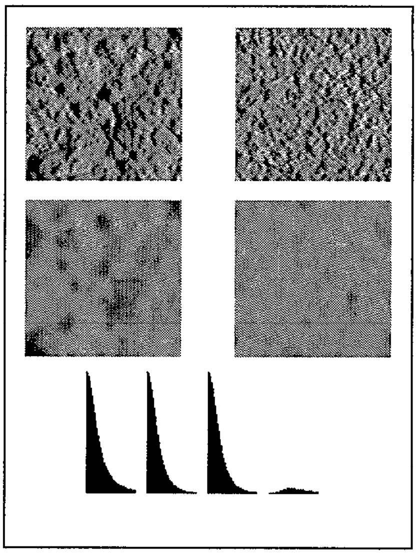


Figure 7.24: Textures 30 and 49.Textures 30 and 49 after Gaussian filtering. Texture scale adjustment using (7.6) and error.

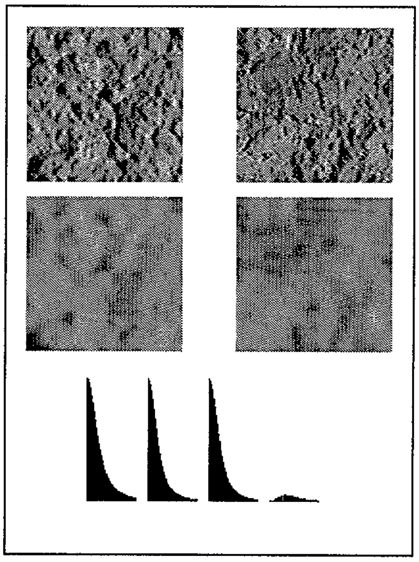


Figure 7.25: Textures 30 and 50.Textures 30 and 50 after Gaussian filtering. Texture scale adjustment using (7.6) and error.

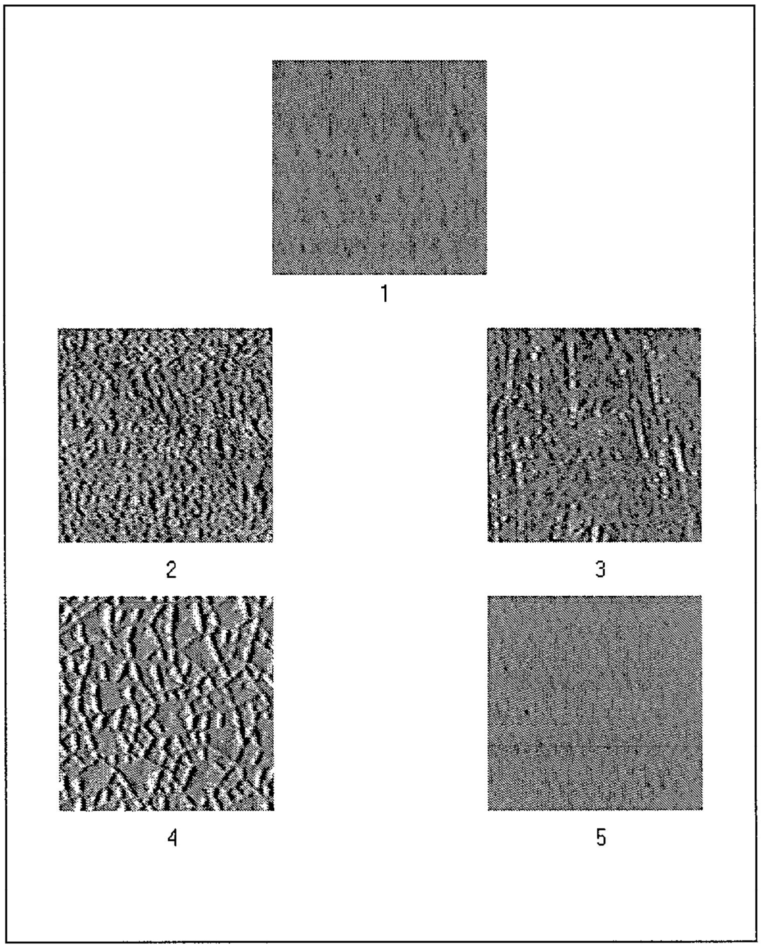


Figure 7.26: Example of the results of searching for similar textures for texture 31 using (7.3).

comparison	energy	weighted energy	quadratic error (*100)	absolute error
31-26	6.05	1.52	3.4939	0.966489
31-27	8.28	2.05	1.8701	0.712228
31-04	7.98	2.11	0.4681	0.373229
31-12	6.64	2.24	0.9072	0.507692

Table 7.11: Comparison of texture 31 with the most similar ones.

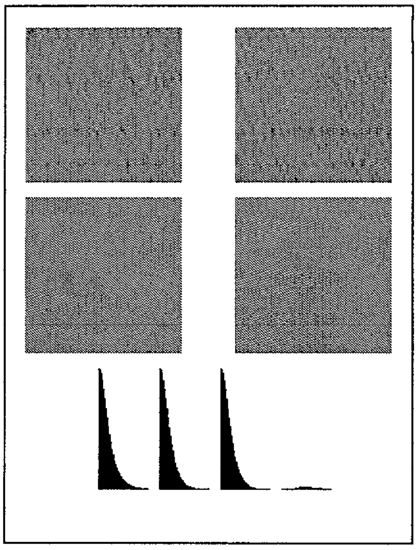


Figure 7.27: Textures 31 and 12.Textures 31 and 12 after Gaussian filtering. Texture scale adjustment using (7.6) and error.

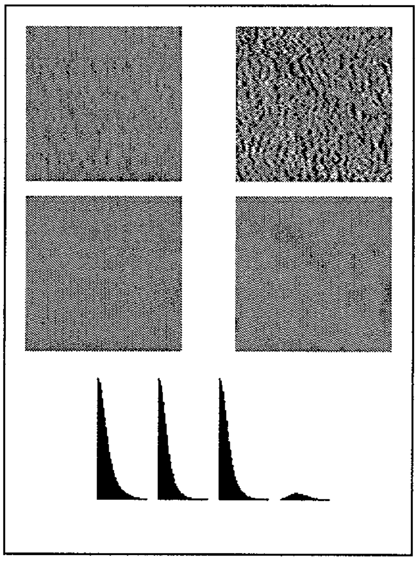


Figure 7.28: Textures 31 and 12.Textures 31 and 12 after Gaussian filtering. Texture scale adjustment using (7.6) and error.

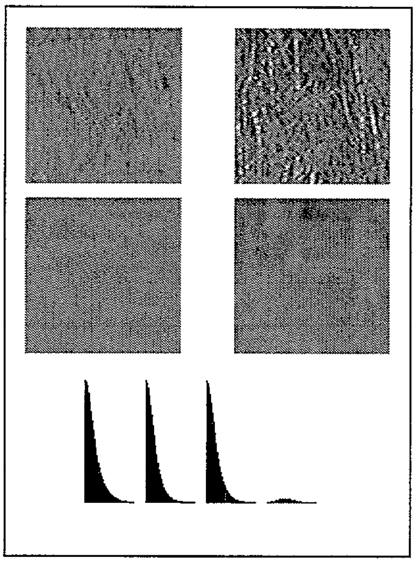


Figure 7.29: Textures 31 and 27.Textures 31 and 27 after Gaussian filtering. Texture scale adjustment using (7.6) and error.

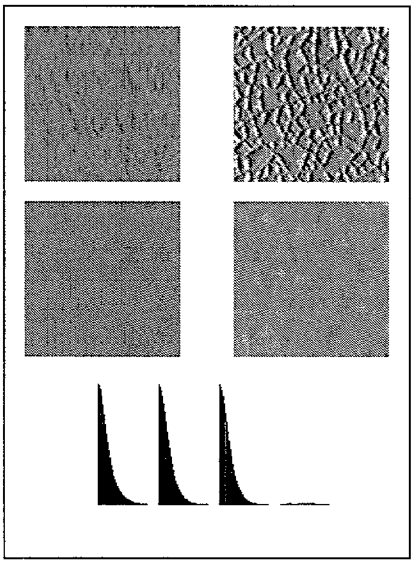


Figure 7.30: Textures 31 and 4.Textures 31 and 4 after Gaussian filtering. Texture scale adjustment using (7.6) and error.

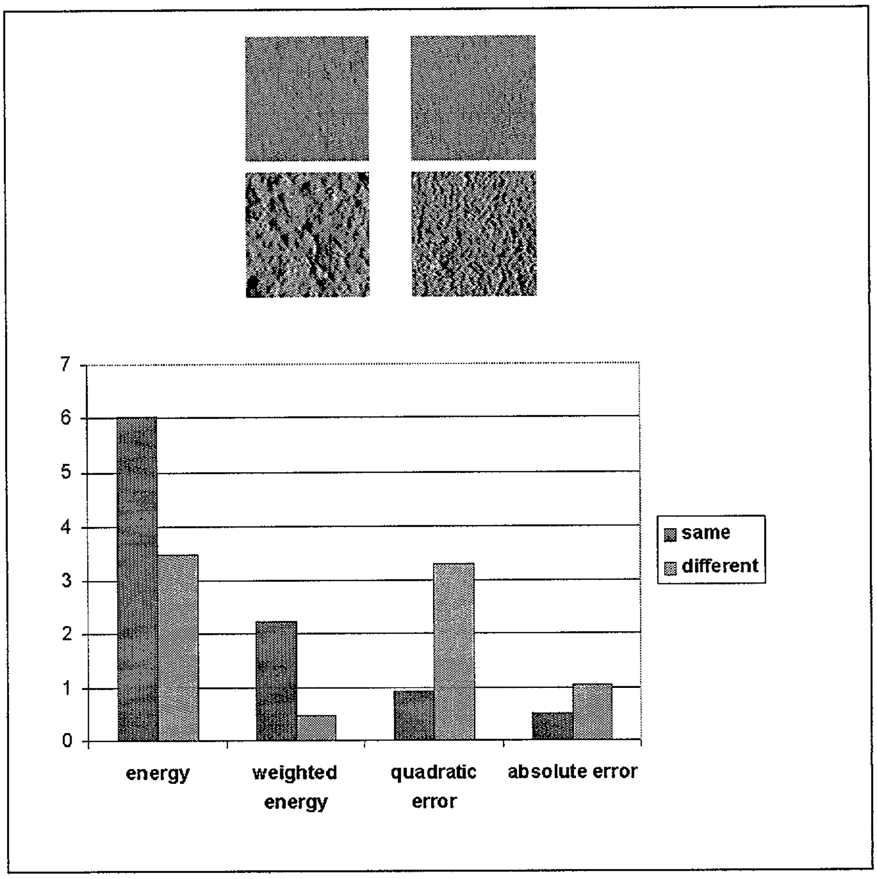


Figure 7.31: Comparison between the energy (7.2), weighted energy (7.3), quadratic error (7.5) and absolute error (7.7) when comparing two images of the same texture at different scales (first couple) and two images of different but similar textures (second couple).

7.3.3 Multiscale texture orientation histogram comparison

We can study how the energy obtained when comparing the orientation histograms evolves as we apply a Gaussian filtering on the textures. When comparing two textures I and I', we firstly calculate the scales where the addition of the squares of the gradients has been reduced to a half of the initial addition, σ and σ', respectively. We use the the adjusting factor k, as in equation (7.6), to calculate an intermediate value σ_N as:

$$\sigma_N = \frac{\sigma + \frac{1}{k}\sigma'}{2}$$

An finally, we obtain the energies for the comparison of the histograms at N different scales, given by equation (7.8), where $n = \{0, .., N-1\}$. With this expressions, after computing the energies at the corresponding N scales, the last comparison analyzes each texture at a scale where the addition of the squares of the gradients is around the half of the initial addition for that texture, and the intermediate comparisons correspond to couples of scales at which both textures are comparable.

$$\begin{aligned} \sigma_n &= \frac{n}{N}\sigma_N \\ \sigma'_n &= \frac{nk}{N}\sigma_N \end{aligned} \tag{7.8}$$

By analyzing the resulting energies with regard to the initial one, we can determine how similar both textures are. For texture 31, if we compare the energies obtained with textures 26, 27, 4 and 12, which were the ones selected using the initial weighted energy, we achieve the results in figures 7.32 and 7.33. Figure 7.32 shows the absolute values of the energy, while figure 7.33 shows the percentages with respect to the initial energy. The value represented for a certain n is the energy obtained when comparing both texures after a certain Gaussian filter is applied, using σ_n for the first texture and σ'_n for the second one. They show that the most similar texture is texture 12, which in fact is the same texture as texture 31, but at a different scale. As observed, while the energy for the comparison between textures 31 and 12 decreases significantly, those obtained for the comparison of texture 31 with textures 26, 27 and 4 increase (texture 26), remain in similar values (texture 27) or decrease very slowly (texture 4).

This allows us to order the textures according to their similarity with the reference one in a more accurate way, since the information provided by the evolution of the enegies as a Gaussian filtering is applied at different scales complements that offered by simply considering the initial energy, thus endowing our mechanism with a higher robustness.

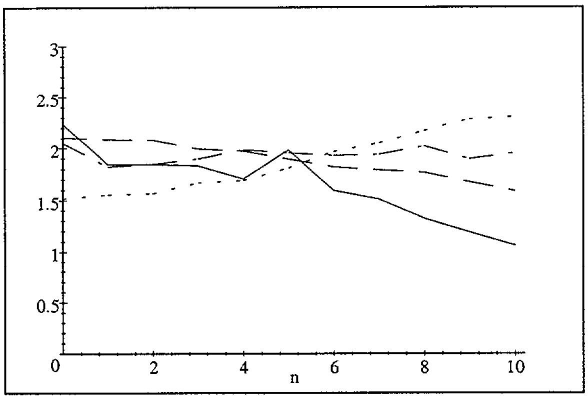


Figure 7.32: Evolution of the energy when applying Gaussian filtering for texture 31 compared with textures 26 (dots), 27 (dot-dash), 4 (dash) and 12 (solid).

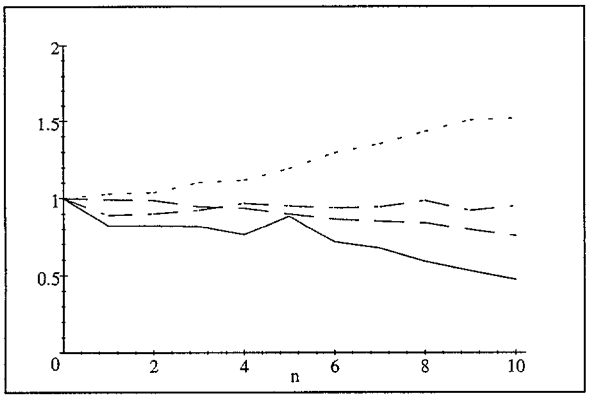


Figure 7.33: Relative evolution of the energy when applying Gaussian filtering for texture 31 compared with textures 26 (dots), 27 (dot-dash), 4 (dash) and 12 (solid).

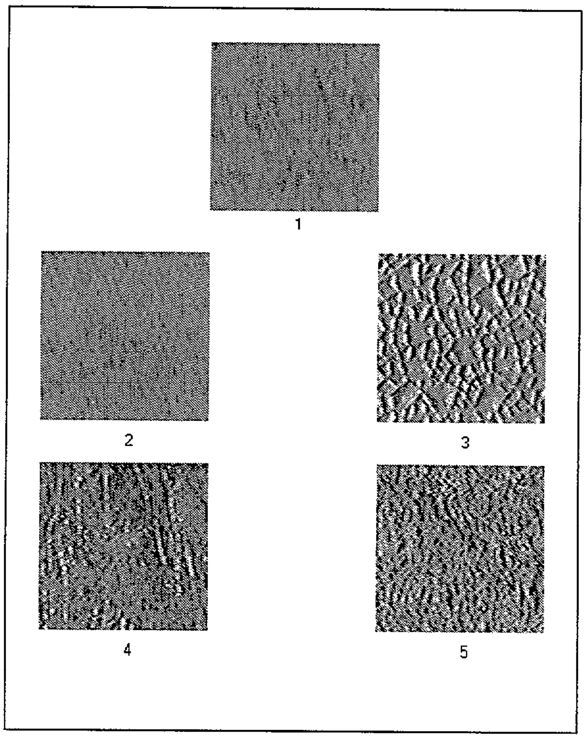


Figure 7.34: Ordered results of comparing texture 31 with textures 12, 4, 27 and 26 using the energy of the different histograms when a Gaussian filtering is applied.

comparison	initial weighted energy	final weighted energy	% final-initial energies
31-12	2.24	1.06	47.42
31-04	2.11	1.59	75.53
31-27	2.05	1.95	95.07
31-26	1.52	2.31	151.96

Table 7.12: Comparison of texture 31 with the most similar ones using the energies at different scales.

7.4 Robustness of texture classification under darkening, lightening and inversion

The following examples show how darkening, lightening and inverting a pattern affect the results when calculating the energy which measures the similarity between two textures. This will allow us to test the robustness of our method when some kinds of transformations are performed in the input signal.

Darkening is obtained by multiplying the original values by a constant, lower than 1. For lightening, the same process is executed, but in this case we use a constant higher than 1. Finally, inversion is obtained by subtracting the input values from 255, which is the maximum value.

As observed in figures 7.35-7.38 and in table 7.13, when a texture is darkened, the resulting energy is very low. This energy measures the dissimilarity between two textures, i.e. the lower the values, the more similar they are. Thus, these low results indicate that the textures are in fact almost identical. Only small differences in the new light intensity due to the representation limitations cause a negligible value. The use of integer values in intensity representation forces us to round the values once they have been reduced, generating small differences in gradient values.

Figures 7.39-7.42 and table 7.14 show similar consequences when the textures are lightened. In this case, the overflow in light intensity values for the most bright points due to the increase they undergo, which forces us to truncate the values which exceed the maximum, causes a higher difference. However, it is still much lower than those observed when similar but different textures are compared, and thus, they can be neglected.

Finally, figures 7.43-7.46 and table 7.15 show the results when a texture has been inverted. In most cases, the values obtained are very low and the patterns can be considered as the same texture. Nevertheless, some cases present certain problems due to the asymmetry of the filters used for edge orientation estimation. As the values which result are not low enough to clarify the similarity of the textures, thus presenting a certain ambiguity, the multiscale analysis described in the previous sections is applied and the results in tables 7.17 and 7.20 dispel the doubts, since they are very low when a texture is compared with its inverted version.

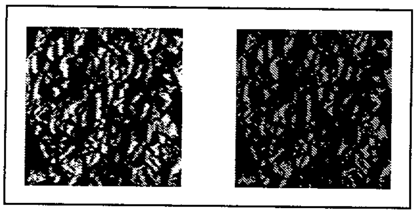


Figure 7.35: Texture 15 before and after darkening.

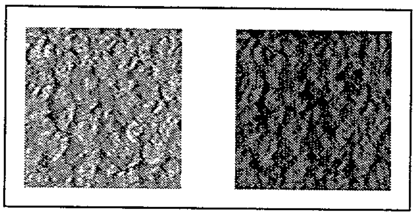


Figure 7.36: Texture 18 before and after darkening.

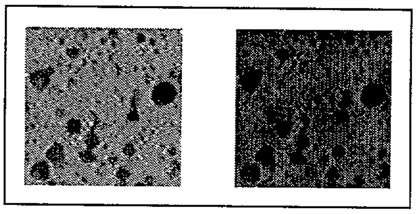


Figure 7.37: Texture 21 before and after darkening.

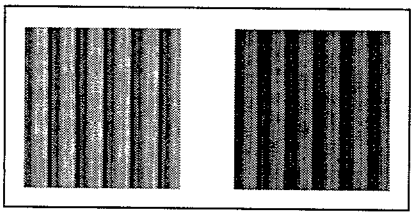


Figure 7.38: Texture 38 before and after darkening.

comparison	weighted energy
15 - dark 15	0.0083
18 - dark18	0.0050
21 - dark 21	0.0129
38 - dark 38	0.0145

Table 7.13: Comparison with dark textures.

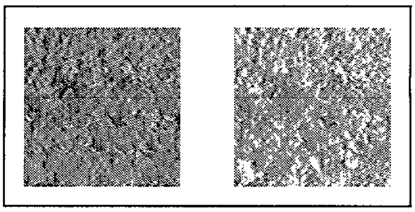


Figure 7.39: Texture 3 before and after lightening.

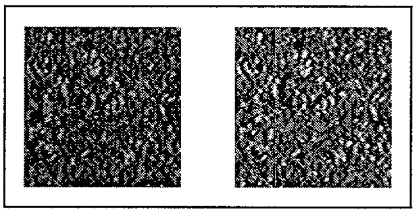


Figure 7.40: Texture 14 before and after lightening.

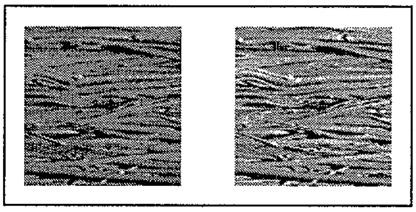


Figure 7.41: Texture 40 before and after lightening.

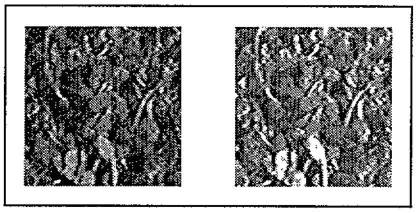


Figure 7.42: Texture 53 before and after lightening.

comparison	weighted energy
03 - light 03	0.0374
14 - light 14	0.0117
40 - light 40	0.0052
53 - light 53	0.0757

Table 7.14: Comparison with light textures.

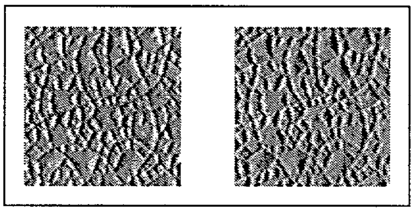


Figure 7.43: Texture 4 before and after inversion.

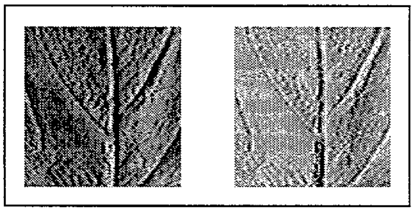


Figure 7.44: Texture 23 before and after inversion.

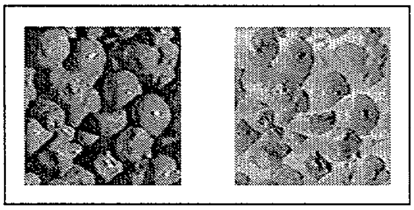


Figure 7.45: Texture 35 before and after inversion.

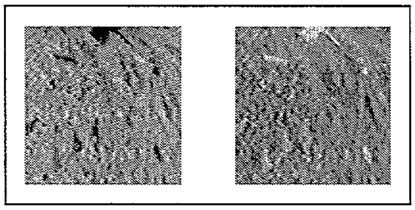


Figure 7.46: Texture 52 before and after inversion.

comparison	weighted energy
04 - inverted 04	0.2567
23 - inverted 23	0.4859
35 - inverted 35	0.6009
52 - inverted 52	0.4927

Table 7.15: Comparison with inverted textures.

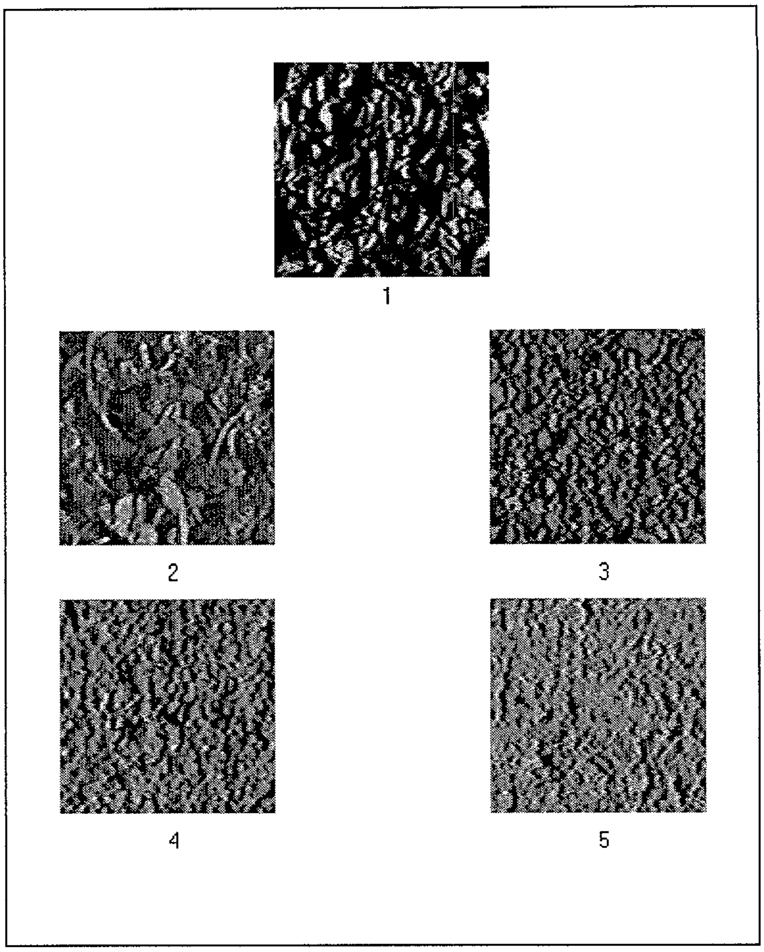


Figure 7.47: Results of searching for similar textures for texture 15 using (7.3).

order	texture number	energy
1	15	0.00
2	53	1.08
3	32	1.43
4	08	2.06
5	49	2.13

Table 7.16: Comparison of texture 15 with the most simlar ones.

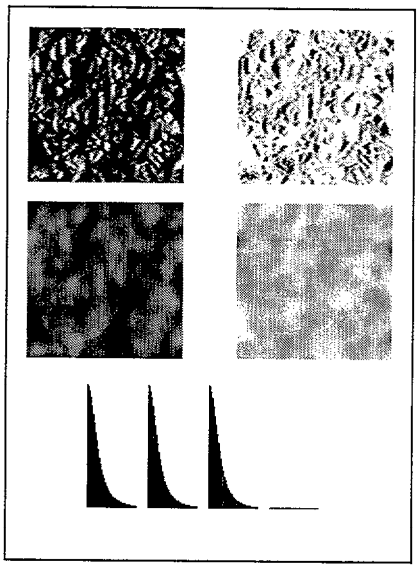


Figure 7.48: Texture 15 before and after inversion. Results of applying a Gaussian filter on texture 15 before and after inversion. Texture scale adjustment using (7.6) and error.

comparison	energy	weighted energy	quadratic error (*100)	absolute error
15- inverted 15	11.8061	3.0951	0.1007	0.170245

Table 7.17: Comparison of texture 15 and its inverse.

comparison	quadratic error (*100)	absolute error
15-53	0.0748	0.140708
15-32	2.0305	0.807616
15-08	1.9196	0.766233
15-49	2.4117	0.873240

Table 7.18: Comparison of texture 15 with the most similar ones.

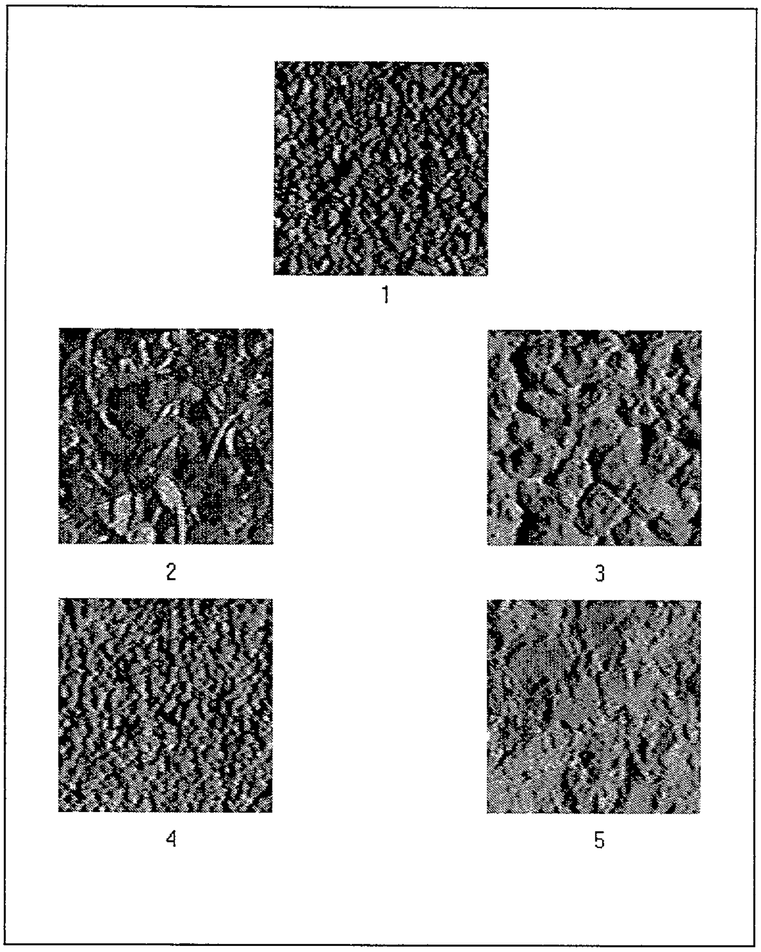


Figure 7.49: Results of searching for similar texture for texture 32 using (7.3).

order	texture number	energy
1	32	0.00
2	53	0.59
3	45	0.91
4	08	1.13
5	50	1.20

Table 7.19: Comparison of texture 32 with the most similar ones.

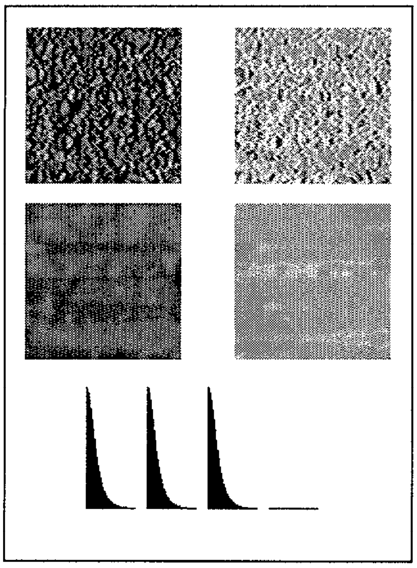


Figure 7.50: Texture 32 before and after inversion. Results of applying a Gaussian filter on texture 32 before and after inversion. Texture scale adjustment using (7.6) and error.

comparison	energy	weighted energy	quadratic error (*100)	absolute error
32- inverted 32	8.0430	2.2133	0.0769	0.129451

Table 7.20: Comparison of texture 32 with its inverse.

comparison	quadratic error (*100)	absolute error
32-53	1.5867	0.713517
32-45	4.5237	1.172753
32-08	0.0440	0.091917
32-50	0.9818	0.562147

Table 7.21: Comparison of tetxure 32 with the most similar ones.

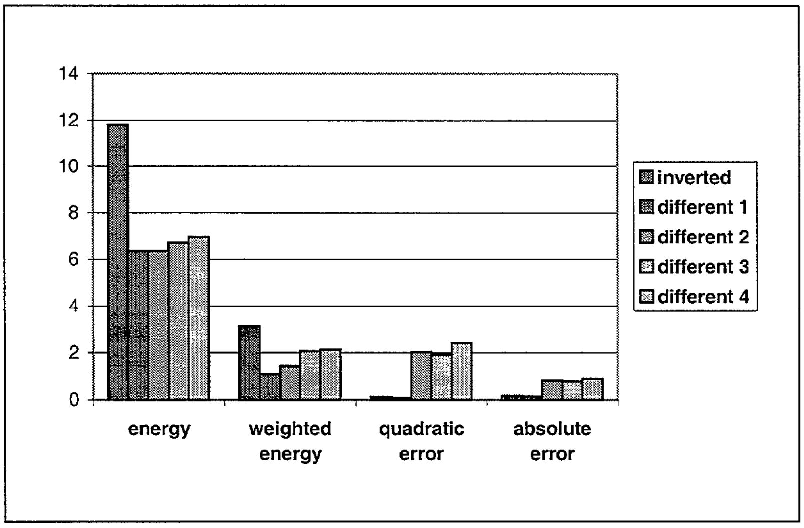


Figure 7.51: Comparison between the energy (7.2), weighted energy (7.3), quadratic error (7.5) and absolute error (7.7) when comparing a texture with its inverted version and the same texture with 4 similar ones.

Chapter 8

COLOR PERCEPTION

Up to now, the processes described in this work deal with greyscale images, i.e. monochromatic information. In that case, the only important information about a certain pixel in the image is its intensity, but not the wavelength of the light it reflects.

For certain tasks such as shape characterization or motion analysis, the color of the objects is not critical in most cases. However, for the global processing of a scene, color plays an important role and the information contained in light intensity is not enough, thus requiring to use hue values. In this chapter we show a brief description of color processing and enhancement as well as the use of orientation filters in image restoration. Furthermore, we explain how color information could be used in processes such as texture classification.

8.1 Equalization

There are different processes to allow a better identification of the elements that are present in an image. Among them, we find equalization. Normally, it is carried out as a global process, but we apply a local process in which the transformations are based on statistical parameters of the region where every pixel (photoreceptor) is located.

Traditional equalization tries to transform the histogram of an image in an ordered way to adapt it to a prefixed desired form, for example, a uniform distribution of the intensities. Such a transformation allows lightening dark images, darkening light images or discriminating objects whose contrast is very low. The computation of this transformation requires the study of the whole image and, if we performed it in three different channels for each type of receptor, the number of operations required would be multiplied. That could be analogous to say that every cell should be affected by the response of the rest of the retina to adapt its output. However, the clearness with which a certain pixel is perceived depends on its contrast with the values in the surrounding area, and not with the whole image.

Due to the inconvenience of the global transformation, we perform a local study of the histogram that would not be the whole histogram of the image, but the one corresponding to a reduced part of it around the pixel to be considered. This information, besides being much more reduced, refers only to the area where the pixel is located, allowing a faster process and a higher contrast enhancement.

However, if the region is not very large, the number of pixels could be smaller than the number of colors to be used and, to avoid a restriction in the range, we use a statistical distribution of the levels in such a way that the mean of the image will be located in the mean of our vision capabilities and the mean deviation will be such that it allows a clear differentiation of the neighbor areas. In order to achieve this goal, each computational unit calculates the mean of the levels of the region where it is located, as well as the mean deviation of these values from the mean. After that, only its own color levels would be transformed according to the following expressions:

$$R' = \frac{(R - m_r)\,S}{s_r} + M$$

$$G' = \frac{(G - m_g)\,S}{s_g} + M$$

$$B' = \frac{(B - m_b)\,S}{s_b} + M$$

where R, G and B are the original values of each channel, R', G' and B' are the transformed ones, m_r, m_g and m_b are the means of the area, M the desired mean, s_r, s_g and s_b are the real deviations and S the desired deviation.

The fact that we are taking into account only statistical values of the area makes it possible to consider only some points inside it, and not the whole number of pixels. In addition, we can adjust the desired mean and deviation, so that we can select the range we want to expand or contract.

On a greyscale image, equalization would be performed on a channel. However, on a color image, we have three channels, which could be equalized in different processes or in a single one. In the first case, we would have three images that would be combined to build the final one, while in the second option we would have only one. With the first option we can enhance the colors that are weaker, but it represents a problem if we try to keep the importance of each channel, as that would yield a false color image, and the hue may be deeply altered. In the second one, the order is kept since the total light works as reference value, and all channels are equally affected.

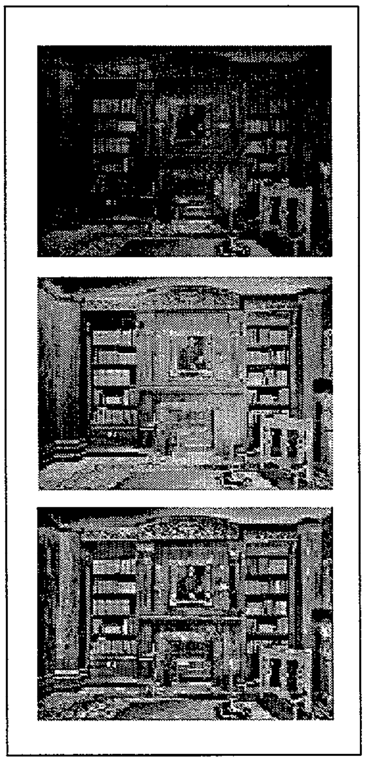


Figure 8.1: Original image, equalized image and enhanced image (1).

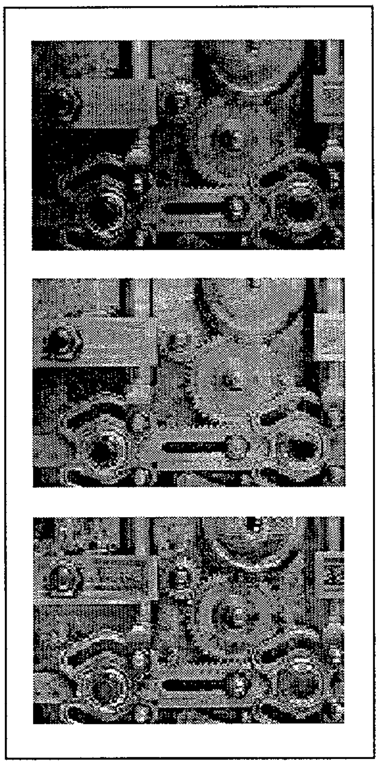


Figure 8.2: Original image, equalized image and enhanced image (2).

8.2 Color enhancement

Different circumstances in image acquisition may produce poorly contrasted pictures which require an enhancement process to be analyzed properly.

8.2.1 Lateral inhibition and color enhancement

In our retina, we find cones that are specialized in different wavelength, but the identification of those which are not exactly their preferred ones is obtained by the interaction of their responses, as shown in figure 8.3. Thanks to the process of lateral inhibition, we can obtain a much more contrasted image in which predominant colors are enhanced. A possible risk of the equalization is that, even if we can expand the histogram, we can lose the hue.

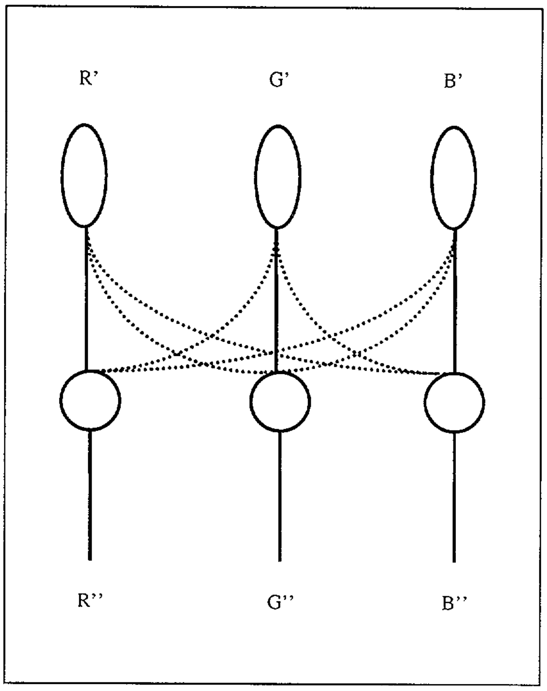


Figure 8.3: Color lateral inhibition.

If blue cones inhibit red and green ones, and vice versa, we can enhance predominant colors. In a matrix expression, we can see it as follows, being R', G' and B' the previous values and $R"$, $G"$ and $B"$ the enhanced ones:

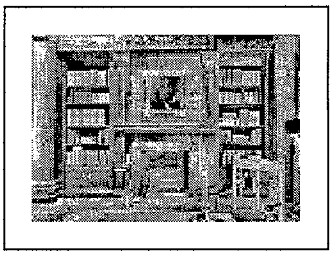


Figure 8.4: Locally equalized image (1).

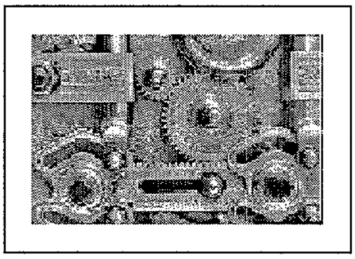


Figure 8.5: Locally equalized image (2).

$$\begin{pmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{pmatrix} \begin{pmatrix} R' \\ G' \\ B' \end{pmatrix} = \begin{pmatrix} R'' \\ G'' \\ B'' \end{pmatrix}$$

If we are not interested in enhancing a color more than the others, a_{ii} would all be equal, and all a_{ij} (i and j different) too. Besides, in order to avoid distortions, $a_{ii} + 2a_{ij} = 1$ with $i \neq j$. In a normal enhancement, the elements on the diagonal would be positive and greater than 1, and out of the diagonal they would be negative. The more different from 1 the elements of the diagonal are and from 0 those out of the diagonal, the higher the enhancement.

The limitation of values to be used makes it necessary to truncate the values when they exceed the limits.

8.2.2 Edge orientation and color enhancement

Since we have developed a mechanism to extract the orientation of the edges, we can use it to determine in which direction the inhibition must take place when enhancement is to be performed. This will allow us to equalize the images locally and to enhance the dominant colors in every region.

From the estimations of gradient orientation and magnitude, the most suitable direction and strength for the diffusion or contrast process can be extracted, in such a way that the edges are preserved or even enhanced while homogeneous regions which may be noisy are homogenized.

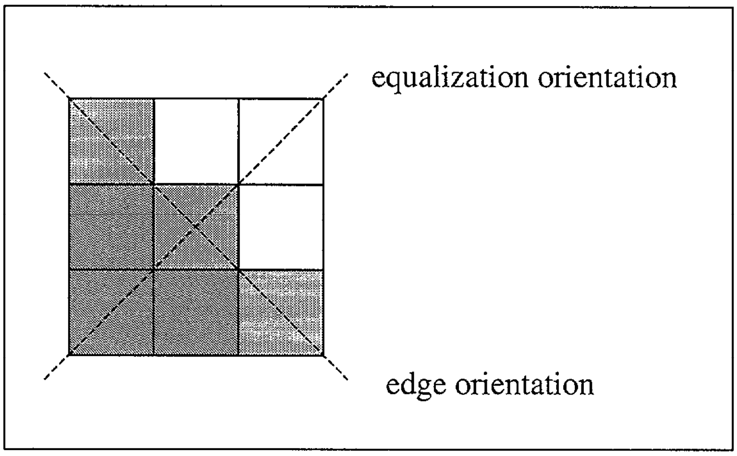


Figure 8.6: Edge orientation and equalization direction.

8.3 The role of color in shape, motion and texture analysis

In the previous chapters, shape, motion and texture have been processed taking into account the light intensity of the points in the image, but not their hue. For many applications it is enough, but for many others, it is important to consider the color of the different parts. Furthermore, it can be very helpful when qualifying objects, discriminating textures or enhancing the contrast between different objects or an object and the background.

For instance, when classifying textures, an orientation histogram has been used to describe the distribution of the different orientation extracted from contour direction estimation. However, if we work with color images, three different histograms could be extracted, thus providing three times as much information as considering a single intensity value.

Chapter 9

CONCLUSION

Many topics on artificial vision and image processing have been included in this work, and after having presented them in detail, this chapter offers the main contributions, their originality, their limitations and the future works which could arise from the aspects studied here.

We propose a unified framework to study different problems in computer vision, like shape representation, motion analysis or texture discrimination, based on contour orientation, which provides quite robust and natural information for the visual system.

The new approaches which have been explained represent a reduced but promising contribution to the development of new techniques and mechanisms for shape characterization, motion analysis and texture classification.

Some aspects have been left untreated, but the basis for many of them have already been established, so that a considerable work could be done in the future for the application of these structures, tools and methods to various fields. Furthermore, the already explained analysis could be adapted to certain particular applications in which shape, motion or texture are critical aspects to take into account.

9.1 Advantages, originality and limitations

The present work proposes some new methods for shape characterization, motion analysis and texture classification based on a new set of tools for characterizing edges. These are inspired on Newton Filters and allow an accurate processing of local information to build a global representation of a shape. Due to the limitations and disadvantages of Newton filters, some modifications have been carried out on them in order to provide them with rotational invariance, but preserving their invariance to global illumination changes. Some other sets of filters have been proposed, such as those described by Sobel, Kirsch, Robinson and Prewitt [SHB99], but we have adapted the features of our filters to the needs of the processes we were to perform. The goal is not the extraction of edge location, but a local estimation of its orientation and magnitude.

The invariance to global illumination changes is achieved by means of a normalization. On the other hand, the rotational invariance is obtained by using cyclic weights in the configuration of the set of filters. This allows comparing the outputs with a pattern to characterize the shapes, in such a way that we identify edge orientation in an accurate way, which is very important when studying curvature, singular points or selective motion detection of an object. This is also improved by means of an interpolation of the values of the pattern provided by the filters in order to find a more accurate estimation. Another interesting property of this new set of filters is that it provides a complete representation of the image border up to a translation grey-level transformation.

From this kind of basic filters, it is possible to build orientation functions which can clearly identify shapes and extract global information to fit objects into described patterns. We have used the discrete Fourier transform as fundamental tool in our anlaysis. The use of Fourier coefficients provides robust results and allows reducing the computational cost. A similar mechanism for shape identification, based on continuous Fourier series, is used in Zahn and Roskies [ZR72], but oriented to polygonal shapes with irregularly spaced series of points, while in our case, signals are discrete, points are equally separated and the fast Fourier transform is the basic tool for the shape analysis. Moreover, the introduction of a weighting function which affects the contribution of every term in the energy function allows regulating the frequencies which will be more relevant in the discrimination.

The way computations are performed and the simplicity of the basic units, by means of which the more complex operations are built, allow a parallel implementation and a layered structure. This makes this set of filters suitable as a retinal computation model. Even more, the completeness of this set of filters is an important aspect concerning the preservation of information if we consider the location and characterization of changes as the main goal.

The information extracted for a segment of an orientation function permits relating different parts of a sequence in such a way that we can associate the visible regions of an object when it is not completely accessible. This requires a common analysis of the different segments which have been extracted, since some conditions must be set in order to guarantee that they do belong to the same shape. Hence, the translation, rotation and scaling relationships must be the same for all couples of associated segments.

Furthermore, the fact that the information supplied by these mechanisms is larger and more accurate than that provided by simply detecting edges allows us to use them for a reliable motion analysis. In this case, the contours which are associated do not belong to different scenes, but to different instants in a video. Once the shapes have been analyzed and coupled, the transformations which bring one of them to the other make it possible to extract motion parameters such as translation, rotation and scaling in time.

Some complementary applications are directly extracted from the properties of these filters. For example, texture analysis and classification are easily implemented by means of a comparison of orientation histograms. In this case, we do not analyze the outline of an object, but the contour in every point of the picture, since what characterizes a texture is the pattern contained inside it. Thanks to the accurate estimation of gradient orientation, we can determine the distribution of the different orientations inside an area, and with similar techniques as those used for orientation functions, we can establish a relationship for orientation histograms. The non-injectivity of this histogram generation, i.e. quite different textures may generate similar orientation histograms, makes it necessary to use some other tool to enhance discrimination. The introduction of multiscale analysis provides a powerful tool for this purpose, since, even if the original orientation histograms may be very similar for clearly different textures, their evolution as a Gaussian filtering is applied will differ, thus allowing to distinguish them. Moreover, it also allows determining the scale relationship of two images obtained from the same texture, but at different resolutions.

The previous features show the usefulness of these filters when the objective is not merely the location of edges, but the discrimination of their orientation. The combination of the local information to build shape representations provides us with a powerful tool for pattern recognition and motion analysis, and the study of the distribution of the different orientations across a region in a multiscale framework permits analyzing textures. The numerical experiences are very promising and show a great coherence with the similarity measures and associations that humans use. In particular, we can even discriminate between shapes which are very similar from a perceptual point of view, follow the trajectory of a moving object and relate visually similar textures.

The examples shown in chapter 5, where a large database containing pictures of 1000 fishes is studied to extract the most similar contours for a given one, produce very promising numerical results, since the selection is very similar to what a human observer would do.

In chapter 6, the numerical experiences with moving objects prove the efficiency of the mechanism when their evolution can be defined in terms of similarity transformations which preserve the outer form.

In chapter 7, the combination of the orientation estimation and the multiscale analysis appears as a powerful technique for texture classification, with energy values which allow a clear clustering of the different types of patterns.

Of course, there are also limitations in all these aspects. Shape characterization has been applied in two dimensions, which means that only a single view of the object is visible. That means that when the image of the object is acquired from a different point of view, the contour may be different and the association may fail. Moreover, objects are supposed to be determined by a single contour. Even if occlusion has been treated, objects with more than one contour, f.i. with holes, have not been handled.

As regards motion analysis, the fact that it is based on two-dimensional shape characterization also makes it dependent of the point of view and the position of the object.

Finally, regarding texture classification, as mentioned in chapter 7, some alterations in the greyscale values may alter the histogram generated by the texture. Even though, the similarity measures and the multiscale analysis which have been applied are robust enough to guarantee a quite trustful classification.

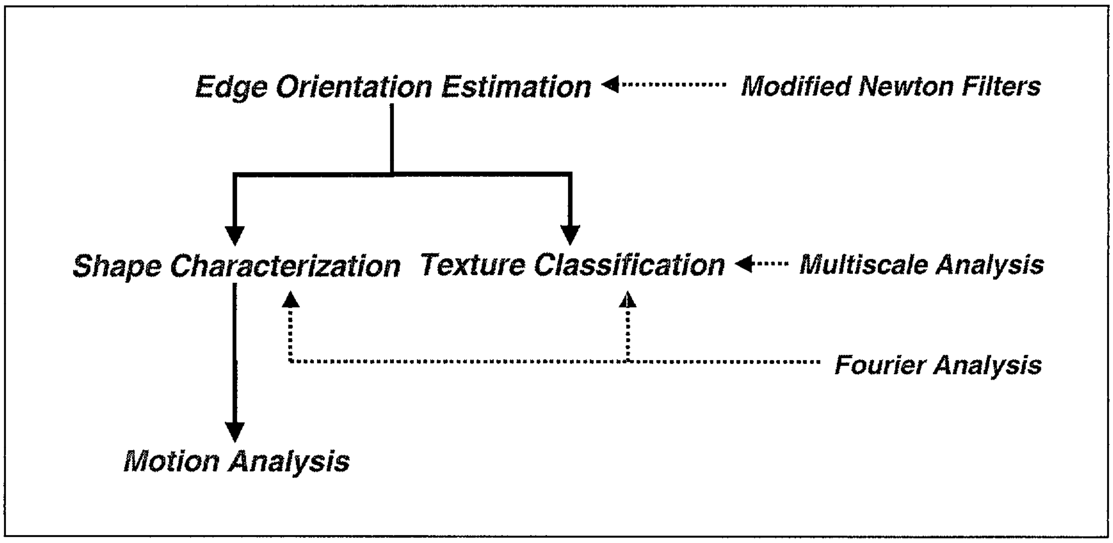


Figure 9.1: General scheme of the work.

9.2 Future work

Due to limitations which have been previously exposed, as well as those aspects which have not been handled, we propose some future works to continue the contribution presented in this work.

In the future, it would be interesting to combine color processing with shape characterization, motion analysis and texture classification in such a way that the information provided by color selectivity makes these tasks more robust and more specific. As explained in chapter 8, the use of three-chromatic systems avoids confusing different objects or object and background when brightness is very similar. Thus, the use of color information may help distinguish regions with very similar intensity values, but with different hue values. The filters to determine edge orientation could be applied to three images, one for each color channel (red, green and blue), and that would endow us with a more reliable contrast information.

The use of orientation selective filters for segmentation, which has not been tackled here, or image enhancement, diffusion and similar processes would enlarge the fields in which the importance of selective units is critical. Due to the fact that we not only locate the edges, but also estimate their orientation, a robust technique for segmentation may arise. When enhancing or diffusing images, it is sometimes necessary to guarantee some features of the initial image, thus requiring a selective process in which the local properties of the image are taken into account. This could be provided by the estimations described in this work.

Some problems have not been grappled with here, such as the characterization of multiple contour objects, e.g. objects with holes. This would require a multiple description of a shape, since more than one orientation function would be necessary to characterize all the contours.

The extension of these techniques to three-dimensional images represent a possibility to enlarge the fields in which they can be applied. Two approaches could be possible. On the one hand, modified Newton filters could be adapted to three dimensions by means of cube-shaped filters with the same philosophy as two-dimensional ones. They could be directly applied to three-dimensional images. On the other hand, two-dimensional filters could be applied to different views of three-dimensional images, and the combination of the information provided by the different views would allow a three-dimensional characterization of a shape. Of course, more complex energy functions would be required to assure the invariance when rotations and translations can be performed in three dimensions.

The use of local orientation values for the points of an object would allow a simpler correspondence between the points in two images of the same object acquired from different cameras. For example, in order to calculate the depth component when stereo vision is used, the points in the first image must be related to the corresponding ones in the second image. The search would be reduced if orientation is considered, and the computational cost of correlation would be deeply reduced, as the possibilities decrease.

From the implementation's point of view, it would be very interesting to study the optimization of the parallelization of the operations by software as well as by hardware, since both, filter processing and Fourier analysis allow a high degree of parallelism. This would result in a more effective calculation and a more realistic simulation of neural computation.

Bibliography

[AAM01] M. Alemán-Flores, L. Álvarez-León and R. Moreno-Díaz jr. *Modified Newton filters for edge orientation estimation, shape representation and motion analysis.* Cuadernos del Instituto Universitario de Ciencias y Tecnologías Cibernéticas, Universidad de Las Palmas de Gran Canaria, 2001; 17: 1-30.

[AGLM93] L. Álvarez, F. Guichard, P.L. Lions y J.M. Morel. *Axioms and fundamental equations of image processing.* Arch. for Rational Mechanics, 1993; 123 (3): 199-257.

[Ale86] S.T. Alexander. *Adaptive signal processing.* Springer Verlag, 1986.

[ALM97] M. Alemán-Flores, K.N. Leibovic and R. Moreno-Díaz jr. *A computational model for visual size, location and movement.* Lecture Notes in Computer Science, Springer Verlag, 1997; 1333: 406-419.

[AM94] L. Álvarez and L. Mazorra. *Signal and image restoration using shock filter and anisotropic diffusion.* SIAM J. on Numerical Analysis, 1994; 31 (2): 590-605.

[BFBB92] J.L. Barron, D.J. Fleet, S.S. Beauchemin and T.A. Burkitt. *Performance of optical flow techniques.* In proceedings, 1992 Computer vision and pattern recognition, Champaign, IL, 1992: 236-242.

[BHL64] H.B. Barlow, R.M. Hill, W.R. Levick. *Retinal ganglion cells responding selectively to direction and speed of image motion in the rabbit.* Journal of Physiology, 1964; 173: 377-407.

[BM87] B. Bhanu and J.C. Ming: *Recognition of occluded objects: a cluster-structure algorithm.* IEEE Transactions on Pattern Analysis and Machine Intelligence, 1987; 20 (2): 199-211.

[Col01] Columbia University and Utrecht University. Columbia-Utrecht Reflectance and Texture Database. *http://www.cs.columbia.edu/CAVE/curet/.index.html.*

[ELJT90] G. Eichmann, C. Lu, M. Jankowski and R. Tolimieri. *Shape representation by Gabor expansion.* SPIE, 1990; 1297. Hybrid Image and Signal Processing II.

[GK90] C.C. Gotlieb and H.E. Kreyszig. *Texture descriptors based on co-occurrence matrices.* Computer Visio, Graphics, and Image Processing, 1990; 51 (1):70-86.

[HR01] U. Hahn and M. Ramscar. *Similarity and Categorization.* Oxford University Press, 2001.

[Hub88] D.H. Hubel. *Eye, brain, and vision.* Freeman, 1988.

[HW62] D.H. Hubel and T.N. Wiesel: *Receptive fields, binocular interaction and functional architecture in the cat's visual cortex.* J. Physiology 1962; 160: 106-154.

[JKS95] R. Jain, R. Katsuri and B.G. Schunck. *Machine Vision.* McGraw-Hill, New York, 1995.

[Kan81] E.R. Kandel. *Processing of form and movement in the visual system.* In Principles of Neural Science, Elsevier, 1981; 366-383.

[Koe78] A. Koestler. *Janus, a summing up.* Random House New York, 1978.

[KM89] N. Kyriati and D. Maydan: *Calcultaing geometrical properties from Fourier representation.* Pattern Recognition, 1989; 22 (5): 469-475.

[KYS96] J.H. Kim, S.H. Yoon and K.H. Sohn: *A robust boundary-based object recognition in occlusion environment by hybrid Hopfield neural networks.* Pattern Recognition, 1996; 29 (12): 2047-2060. Pattern Recognition Society, Elsevier Science Ltd.

[Law79] K.I. Laws. *Texture energy measures.* In DARPA Image understanding workshop, Los Angeles, CA, 1979: 47-51.

[Lei66] K.N. Leibovic. *A model for information processing with reference to vision.* J. theore. Biology, 1966; 11: 112-130.

[Lei88] K.N. Leibovic. *Parallel processing in nervous systems with converging and diverging transmission.* Biomathematics and related computational problems. Kluwer Academic Publishers, 1988; 65-72.

[Let62] J.Y. Lettvin. *Form-function relations in neurons.* Research Lab. of Electronics, MIT Quarterly Progress Report, 1962: 333-335.

[LC87] C.C. Lin and R. Chellappa. *Classification of partial 2-D shapes using Fourier descriptors.* IEEE transactions on Pattern Analysis and Machine Intelligence, 1987; 9 (5): 686-690.

[LMMP59] J.Y. Lettvin, H.R. Maturana, W.S. McCulloch and W.H. Pitts. *What the frog's eye tells the frog brain.* 1959.

[Lim90] J.S. Lim. *Two-dimensional signal and image processing.* Prentice Haall, 1990.

[Lin94] T. Lindeberg. *Scale space theory in computer vision.* Kluwer Academic Publishers, 1994.

[Lon98] S. Loncaric. *A survey of shape analysis techniques.* Pattern Recognition, 1998; 31 (8): 983-1001.

[MKN95] K. Miura, K. Kurata and T. Nagano. *Self-organization of the velocity selectivity of a directionally selective neural network.* Biological Cybernetics, Springer Verlag, 1995; 73: 401-407.

[Mok97] F. Mokhtarian. *Silhouette-based occluded object recognition through curvature scale space.* Machine Vision and Applications, Springer Verlag, 1997; 10: 87-97.

[Mok01] F. Mokhtarian. Centre for Vision, Speech, and Signal Processing of the University of Surrey. *http://www.ee.surrey.ac.uk/Research/VSSP/imagedb/demo.html.*

[Mor77] R. Moreno-Díaz. *Models of retinal processes.* Brain Theory Newsletter 1977; 3 (1): 7-10.

[Mor93] R. Moreno-Díaz jr. *Computación paralela y distribuida: relación estructura-función en retinas.* Tesis Doctoral, 1993.

[MQA00] R. Moreno-Díaz jr., A. Quesada-Arencibia and M. Alemán-Flores. *Bases of a pre-attentional mechanism by means of presynaptic inhibition in the lateral geniculate nucleus.* SPIE (The International Society for Optical Engineering), 2000; 3981.

[MQA01] R. Moreno-Díaz jr., A. Quesada-Arencibia and M. Alemán-Flores. *Percepción visual y modelos computacionales. Introducción a la biocibernética visual.* Vicerrectorado de Investigación y Desarrollo Tecnológico. Universidad de Las Palmas de Gran Canaria, 2001.

[MR79] R. Moreno-Díaz, E. Rubio-Royo. *A theoretical model for layered visual processing.* Int. J. Bio-Medical Computing, Springer, 1979; 10: 231-243.

[MRR78] R. Moreno-Díaz, E. Rubio-Royo, F. Rubio-Royo. *A generalized model for non-linear processing.* Current topics in Cybernetics and Systems, Springer, 1978: 202-203.

[OH97] C.F. Olson, D.P. Huttenlocher. *Automatic target recognition by matching oriented edge pixels.* IEEE Transactions on Image Processing, 1997; 6 (1): 103-113.

[PC95] P.N. Prokopowicz and P.R. Cooper. *The dynamic retina: contrast and motion detection for active vision.* International Journal of Computer Vision 1995; 16: 191-204.

[PG90] C.M.A. Pennartz and W.A. van de Grind. *Simulation of movement detection by direction-selective ganglion cells in the rabbit and squirrel retina.* Vision Research 1990; 30 (8): 1223-1234.

[QAM99] A. Quesada-Arencibia, M. Alemán-Flores, R. Moreno-Díaz jr. *Newton Filters: a new class of neuron-like discrete filters and an application to image processing.* Artificial Neural Nets and Genetic Algorithms, Springer Computer Science, 1999; 28-34.

[RCC89] J.H. Rong, J.L. Coatrieux and R. Collorec. *Combining motion estimation and segmentation in digital subtracted angiograms analysis.* In sixth multidimensional signal processing workshop, Pacific Grove, CA, IEEE, 1989.

[Rod96] A. Rodríguez-Rodríguez. *Desarrollo de una Arquitectura CBR para la Resolución de Tareas basada en el Pensamiento Analógico Intradominio.* Tesis Doctoral, 1996.

[RSH96] Y. Rui, A.C. She, T.S. Huang. *Modified Fourier descriptors for shape representation-a practical approach.* In Proceedings of the 1st workshop on image databases and multimedia search, 1996.

[RWW90] T.R. Reed, H. Wechsler and M. Werman. *Texture segmentation using a diffusion region growing technique.* Pattern Recognition, 1990; 23 (9):953-960.

[SB84] M. Shridhar and A. Badreldin. *High accuracy charcter recognition algorithm using Fourier and topological descriptors.* Pattern Recognition, 1984; 17 (5): 515-524.

[SD92] L.H. Staib and J.S. Duncan: *Boundary finding with parametrically deformable models.* IEEE Transactions on Pattern Analysis and Machine Intelligence, 1992; 14 (11): 1061-1075.

[SHB99] M. Sonka, V. Hlavac, R. Boyle. *Image processing, analysis, and machine vision.* PWS-ITP, 1999.

[Sla90] M. Slaughter. *The vertebrate retina.* In Science of Vision, K.N. Leibovic (ed.), Springer Verlag, 1990.

[Sob74] I.M. Sobol. *Lecciones populares de matemáticas. Método de Montecarlo.* Editorial Nauka-Mir, 1974.

[SU91] A. Shashua and S. Ullman. *Grouping contours by iterated pairing network.* Advances in neural information processing systems. Morgan Kaufmann publishers 1991; 3: 335-341.

[TMV85] J.L. Turney, T.N. Mudge and R.A. Volz. *Recognizing partially occluded parts.* IEEE Transactions on Pattern Analysis and Machine Intelligence, 1985; 7: 410-421.

[Wie60] N. Wiener. *Cibernética.* Guadiana, 1960.

[WW80] T.P. Wallace and P.A. Wintz. *An efficient three-dimensional aircraft recognition algorithm using normalized Fourier descriptors.* Computer Graphics and Image Processing, 1980; 13: 99-126.

[YO97] J.W. Yi and J.H. Oh. *Recursive resolving algorithm for multiple stereo and motion matches.* Image and vision computing 1997; 15: 181-196.

[Zar96] M.R. Zargham. *Computer architecture. Single and parallel systems.* Prentice Hall, 1996.

[ZR72] C. Zahn, R. Roskies. *Fourier descriptors for plane closed curves.* Computer Graphics and Image Processing, 1972; 21: 269-281.

Appendix A

MODELOS DE PERCEPCIÓN VISUAL BASADOS EN LA ORIENTACIÓN DE CONTORNOS (Resumen en español)

En este apéndice presentamos un resumen en español del trabajo, en el cual resaltamos los puntos más significativos de la investigación realizada. Tras una explicación del ámbito de estudio y las bases sobre las que se cimenta, explicamos las aportaciones principales y algunos ejemplos de los resultados obtenidos. Finalmente presentamos las conclusiones más relevantes y ofrecemos algunas líneas de investigación futuras derivables de este trabajo.

A.1 Objeto y objetivos de la investigación

Debido a la importancia de la visión en la interacción de los animales con su entorno, una gran proporción de los esfuerzos humanos por construir máquinas que se comporten como seres vivos ha sido dedicada a la visión artificial. Dentro de este ámbito de conocimiento, hemos desarrollado un marco para el procesamiento de la información visual en varios canales que están especializados en diferentes aspectos del análisis de la información, tales como la forma, el movimiento, la textura y el color, pero que mantienen unos ciertos elementos de cohesión y hacen uso del mismo tipo de herramientas básicas.

La identificación de un objeto está proporcionada, en la mayoría de los casos, por una descripción de su forma, lo cual requiere una localización precisa de sus bordes. Sin embargo, es importante que cambios en la orientación, el tamaño o el contraste entre el objeto y el fondo no alteren la salida del mecanismo usado para el reconocimiento. Este trabajo presenta una descripción multicanal del procesamiento de la información visual en la que, a partir de unas etapas iniciales en las que las características de más bajo nivel son extraídas, diferentes subsistemas analizan la información para propósitos más específicos, tales como la discriminación de la forma, el movimiento, la textura y el color.

Introducimos un conjunto de herramientas formales, basadas en los filtros de Newton, que permiten estimar la orientación de los bordes y cuya salida no varía cuando la señal de entrada es rotada o cuando ocurre un cambio global de iluminación. Además, las operaciones son realizadas en una estructura en capas que simula la actividad de las células ganglionares de los vertebrados superiores. Las salidas de estos filtros son usadas para diferentes propósitos, ya que la detección selectiva de los bordes es una base para otras primitivas de mayor nivel semántico.

Primeramente, permite extraer una representación unidimensional del contorno con el fin de caracterizar formas usando los coeficientes de Fourier. A partir de estos coeficientes, se construye una función de energía para discriminar las formas. Este trabajo presenta un método robusto para la caracterización de objetos que permite discriminar entre formas bastante similares, tales como llaves o peces. El uso de una función de orientación para representar una forma y su análisis a partir de los coeficientes de Fourier hacen posible extraer relaciones para los diferentes modos en que un objeto puede ser presentado, constituyendo así un mecanismo muy útil para muchos propósitos diferentes en áreas como la industria, la medicina, la meteorología, etc.

En segundo lugar, los parámetros usados al comparar los contornos de los objetos permiten estudiar una secuencia de imágenes y extraer información de movimiento, lo cual representa la introducción de la evolución temporal en la escena. La información obtenida para relacionar las formas se usa también para una asociación temporal de imágenes en las cuales un objeto evoluciona. Del mismo modo que la representación de formas está basada en la extracción de la orientación de los bordes en cada punto del contorno, el análisis del movimiento está basado en la representación de formas.

Finalmente, así como la orientación es usada para comparar contornos por medio de un análisis en la frecuencia de las funciones que las caracterizan, las texturas pueden ser estudiadas y clasificadas. Para cada textura, se construye un histograma de orientación insertado en el marco de un análisis multiescala, con el fin de describirla y compararla con otros patrones. En este caso, el análisis multiescala de las texturas proporciona mucha más información, ya que es en la evolución de los patrones a lo largo de las escalas donde las diferencias entre las texturas pueden ser detectadas.

Los filtros usados para estimar la orientación de los bordes constituyen una herramienta simple pero efectiva para estos objetivos y, al mismo tiempo, son apropiados para una estructura en capas como las encontradas en los sistemas naturales. Por otro lado, el análisis de Fourier de las funciones de orientación permite extraer algunas características generales de las formas que son presentadas. Más aún, proponemos el uso de funciones de peso para los coeficientes de las diferentes frecuencias, de tal forma que aquellos coeficientes cuya información es más relevante posean un peso mayor en el esquema resultante. Analizamos las formas de tales funciones de peso con el fin de mejorar la discriminación y reducir la probabilidad de fallo.

Las experiencias numéricas son muy prometedoras. En particular, podemos discriminar incluso entre formas que son muy similares desde un punto de vista perceptual. Todas estas aplicaciones de un mismo conjunto básico de filtros nos proporcionan un marco global para el procesamiento de la información visual, donde la combinación de módulos simples produce la abstracción de representaciones del mundo exterior de mayor nivel semántico.

Así, dentro de los diferentes canales de proceso, podemos distinguir una serie de niveles que, basándose en los niveles inferiores, infieren nuevas primitivas con una mayor abstracción, desde un nivel físico, hasta un nivel motor de reacción, como se muestra en la figura A.1.

Nivel Motor
Nivel Interpretativo
Nivel Analógico Nivel Lógico
Nivel Temporal
Nivel Espacial
Nivel Físico

Table A.1: Esquema propuesto para los niveles de proceso en el camino visual.

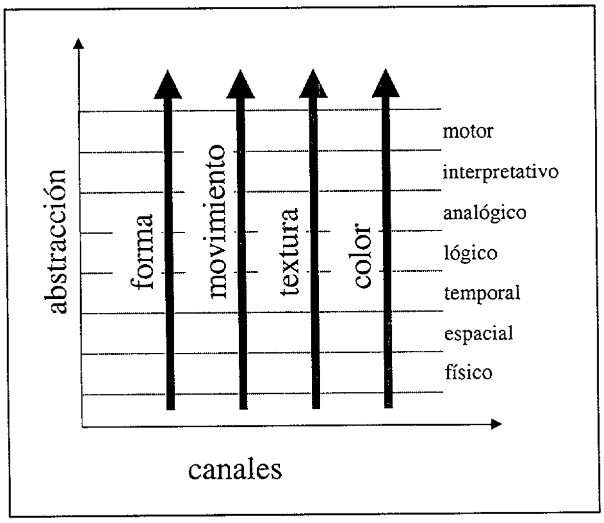


Figure A.1: Representación bidimensional del espacio de transformaciones.

A.2 Planteamiento y metodología utilizada

Como base para el desarrollo posterior de las herramientas de proceso hemos utilizado los filtros de Newton. Los filtros de Newton son estructuras basadas en el uso repetido de operaciones binarias simples para construir otras más complejas que puedan computar un rango más amplio de entradas. Si usamos la suma y la resta de dos números reales o enteros como operaciones básicas, podemos combinarlas en diferentes capas, de forma que las funciones resultantes son combinaciones lineales de los valores de entrada. Cada capa opera sobre el conjunto ordenado de resultados de la capa previa y la operación que se lleva a cabo es la misma para cada unidad dentro de ella, independientemente de la posición donde es ejecutada. Sin embargo, las operaciones de las diferentes capas pueden variar, como se muestra en las figuras A.2 y A.3.

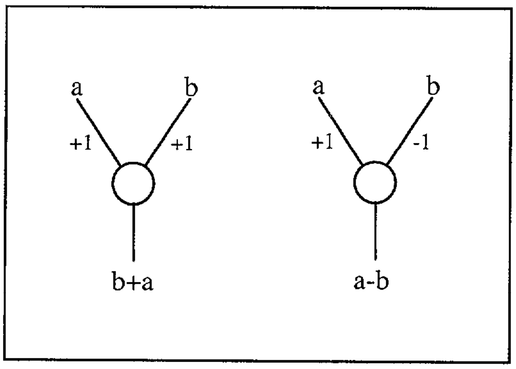


Figure A.2: Operaciones básicas de los filtros de Newton.

$$NF(A_2, D_0) = (1, \ \ 2, \ \ 1)$$

$$NF(A_1, D_1) = (1, \ \ 0, -1)$$

$$NF(A_0, D_2) = (1, -2, \ \ 1)$$

Para una señal de entrada de longitud L podemos construir L filtros linealmente independientes que computen L valores sobre ellos, desde el posee sólo capas aditivas, hasta el que posee sólo capas substractivas. Si representamos este conjunto de filtros en forma de matriz, obtenemos matrices de transformación como la siguiente:

$$Ax = \begin{pmatrix} a_{0,0} & a_{0,1} & \cdots & a_{0,L-1} \\ a_{1,0} & a_{1,1} & \cdots & a_{1,L-1} \\ \cdots & \cdots & \cdots & \cdots \\ a_{L-1,0} & a_{L-1,1} & \cdots & a_{L-1,L-1} \end{pmatrix} \begin{pmatrix} x_0 \\ x_1 \\ \cdots \\ x_{L-1} \end{pmatrix} = \begin{pmatrix} f_0 \\ f_1 \\ \cdots \\ f_{L-1} \end{pmatrix} = F$$

Podemos combinar filtros monodimensionales para construir un nuevo conjunto de estructuras que operen sobre datos bidimensionales. Éstos pueden ser creados considerando la adición y substracción de 4 elementos como operaciones básicas, con la limitación de que el peso global debe ser 0 si existen substracciones. Los 7 filtros obtenidos así, que se muestran en la tabla A.2, no constituyen un conjunto completo, puesto que no es posible recuperar la información original a partir de los resultados.

La multiplicación de filtros horizontales y verticales, lo que equivale a aplicar filtros bidimensionales repetidamente, sí que genera un conjunto completo. Son los que denominamos filtros de Newton expandidos, y se muestran en la tabla A.3.

$$A = \begin{pmatrix} 1 & 2 & 1 & 2 & 4 & 2 & 1 & 2 & 1 \\ 1 & 0 & -1 & 2 & 0 & -2 & 1 & 0 & -1 \\ 1 & 2 & 1 & 0 & 0 & 0 & -1 & -2 & -1 \\ 1 & 0 & -1 & 0 & 0 & 0 & -1 & 0 & 1 \\ 1 & 2 & 1 & -2 & -4 & -2 & 1 & 2 & 1 \\ 1 & -2 & 1 & 2 & -4 & 2 & 1 & -2 & 1 \\ 1 & 0 & -1 & -2 & 0 & 2 & 1 & 0 & -1 \\ 1 & -2 & 1 & 0 & 0 & 0 & -1 & 2 & -1 \\ 1 & -2 & 1 & -2 & 4 & -2 & 1 & -2 & 1 \end{pmatrix}$$

Intentamos construir filtros que reaccionen a cambios en las 8 direcciones principales. Por lo tanto, es necesario utilizar diferentes signos a ambos lados del borde. Sería deseable que los pesos en la región central del filtro no fuesen nulos, como ocurre en los filtros originales, puesto que esto causa la duplicación de los bordes, incluso cuando éstos son perfectos. Para mantener el peso global igual a 0, multiplicamos la región negativa por 2, obteniendo los filtros que se muestran en la figura A.4 y en la tabla A.4.

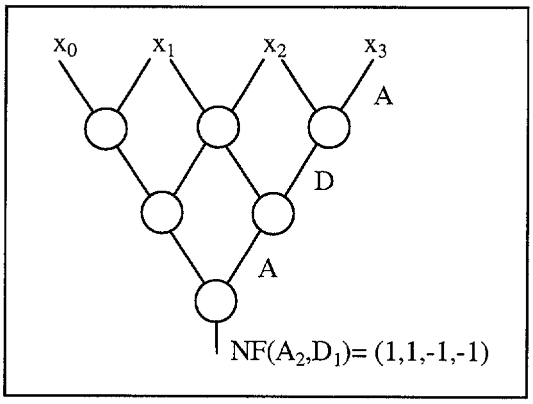


Figure A.3: Ejemplo de un filtro de Newton con dos capas aditivas y una substractiva.

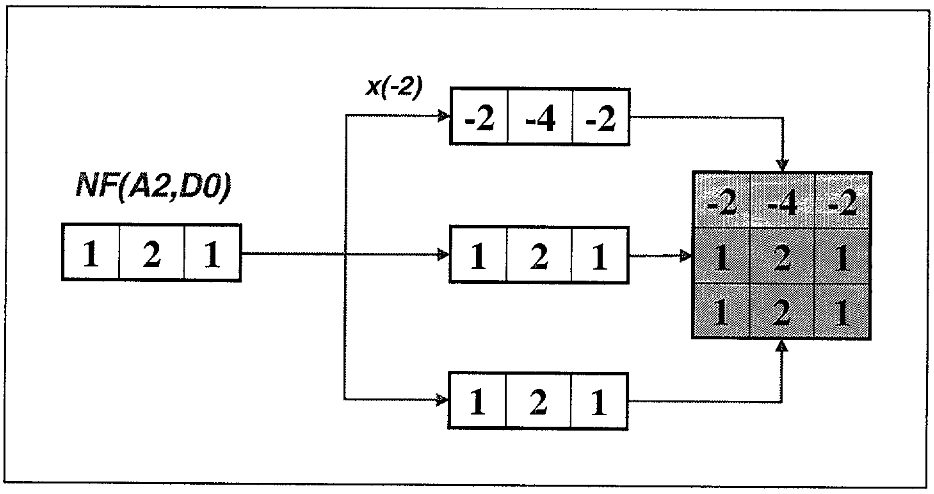


Figure A.4: Proceso de construcción de los filtros de Newton modificados.

$$A = \begin{pmatrix} -4 & -2 & 1 & 1 & 2 & 1 & 1 & -2 \\ -2 & -4 & -2 & 1 & 1 & 2 & 1 & 1 \\ 1 & -2 & -4 & -2 & 1 & 1 & 2 & 1 \\ 1 & 1 & -2 & -4 & -2 & 1 & 1 & 2 \\ 2 & 1 & 1 & -2 & -4 & -2 & 1 & 1 \\ 1 & 2 & 1 & 1 & -2 & -4 & -2 & 1 \\ 1 & 1 & 2 & 1 & 1 & -2 & -4 & -2 \\ -2 & 1 & 1 & 2 & 1 & 1 & -2 & -4 \end{pmatrix}$$

Como se observa en los ejemplos siguientes, cuando la orientación del borde se varía un múltiplo de 45 grados, el patrón es simplemente desplazado cíclicamente, pero los valores dentro de él no son alterados. En la figura A.5, podemos ver la salida de los 8 filtros cuando se aplican sobre un círculo. Cada uno de los filtros reacciona en la región alrededor de su dirección preferida.

1	1	0
1	1	0
1	1	0

Entrada para un borde orientado en 0 radianes.

f_0	f_1	f_2	f_3	f_4	f_5	f_6	f_7
8	5	0	−4	−4	−4	0	5

Salida para un borde orientado en 0 radianes.

0	0	0
1	1	1
1	1	1

Entrada para un borde orientado en $\pi/2$ radianes.

f_0	f_1	f_2	f_3	f_4	f_5	f_6	f_7
0	5	8	5	0	−4	−4	−4

Salida para un borde orientado en $\pi/2$ radianes.

Debido a que la suma de los pesos es cero en todos los filtros, podemos normalizar los valores de salida de manera que un cambio global de iluminación de la forma $I'(x,y) = aI(x,y) + b$, donde I es la señal original e I' es la señal transformada, no produzca una alteración de estos valores:

$$\textstyle\sum_{i,k} w_{i,k}^{n} = 0 \quad i,k \in \{-1,0,1\}$$

$$F_n(x,y) = \sum_{i,k} w_{i,k}^{n} I(x+i, y+k)$$

$$T_{n,m}(x,y) = \frac{F_n(x,y)}{F_m(x,y)}$$

$$I'(x,y) = aI(x,y) + b$$

$$\begin{aligned} T'_{n,m}(x,y) &= \frac{a\sum_{i,k} w_{i,k}^{n} I(x+i,y+k) + b\sum_{i,k} w_{i,k}^{n}}{a\sum_{i,k} w_{i,k}^{m} I(x+i,y+k) + b\sum_{i,k} w_{i,k}^{m}} \\ &= \frac{aF_n(x,y)}{aF_m(x,y)} \end{aligned}$$

$$T'_{n,m}(x,y) = T_{n,m}(x,y)$$

Este trabajo presenta un amplio rango de aplicaciones de los filtros de Newton modificados en diferentes campos del procesamiento de imágenes. Usando el análisis de Fourier, el análisis multiescala, los métodos de minimización de errores y otras herramientas matemáticas, y partiendo de una estimación muy aproximada de la orientación de los bordes, algunas características, tales como la forma de la silueta de un objeto o la trayectoria de un objeto en movimiento pueden ser examinados de forma muy precisa y con resultados numéricos bastante satisfactorios. De forma similar, también es posible analizar y clasificar texturas a partir de histogramas de orientación obtenidos a partir de las salidas de los mismos filtros.

	$\boxed{\begin{matrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{matrix}}$ $N_0 : \begin{pmatrix} + & + \\ + & + \end{pmatrix}$	
$\boxed{\begin{matrix} 1 & 0 & -1 \\ 2 & 0 & -2 \\ 1 & 0 & -1 \end{matrix}}$ $N_1 : \begin{pmatrix} + & - \\ + & - \end{pmatrix}$		$\boxed{\begin{matrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{matrix}}$ $N_2 : \begin{pmatrix} - & + \\ - & + \end{pmatrix}$
$\boxed{\begin{matrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{matrix}}$ $N_3 : \begin{pmatrix} + & + \\ - & - \end{pmatrix}$		$\boxed{\begin{matrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{matrix}}$ $N_4 : \begin{pmatrix} - & - \\ + & + \end{pmatrix}$
$\boxed{\begin{matrix} 1 & 0 & -1 \\ 0 & 0 & 0 \\ -1 & 0 & 1 \end{matrix}}$ $N_5 : \begin{pmatrix} + & - \\ - & + \end{pmatrix}$		$\boxed{\begin{matrix} -1 & 0 & 1 \\ 0 & 0 & 0 \\ 1 & 0 & -1 \end{matrix}}$ $N_6 : \begin{pmatrix} - & + \\ + & - \end{pmatrix}$

Table A.2: Filtros de Newton bidimensionales originales.

	$\begin{array}{\|ccc\|}\hline 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \\ \hline\end{array}$ E_0 : $(1,2,1)^t(1,2,1)$	
$\begin{array}{\|ccc\|}\hline 1 & 0 & -1 \\ 2 & 0 & -2 \\ 1 & 0 & -1 \\ \hline\end{array}$ E_1 : $(1,2,1)^t(1,0,-1)$		$\begin{array}{\|ccc\|}\hline 1 & -2 & 1 \\ 2 & -4 & 2 \\ 1 & -2 & 1 \\ \hline\end{array}$ E_5 : $(1,2,1)^t(1,-2,1)$
$\begin{array}{\|ccc\|}\hline 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \\ \hline\end{array}$ E_2 : $(1,0,-1)^t(1,2,1)$		$\begin{array}{\|ccc\|}\hline 1 & 0 & -1 \\ -2 & 0 & 2 \\ 1 & 0 & -1 \\ \hline\end{array}$ E_6 : $(1,-2,1)^t(1,0,-1)$
$\begin{array}{\|ccc\|}\hline 1 & 0 & -1 \\ 0 & 0 & 0 \\ -1 & 0 & 1 \\ \hline\end{array}$ E_3 : $(1,0,-1)^t(1,0,-1)$		$\begin{array}{\|ccc\|}\hline 1 & -2 & 1 \\ 0 & 0 & 0 \\ -1 & 2 & -1 \\ \hline\end{array}$ E_7 : $(1,0,-1)^t(1,-2,1)$
$\begin{array}{\|ccc\|}\hline 1 & 2 & 1 \\ -2 & -4 & -2 \\ 1 & 2 & 1 \\ \hline\end{array}$ E_4 : $(1,-2,1)^t(1,2,1)$		$\begin{array}{\|ccc\|}\hline 1 & -2 & 1 \\ -2 & 4 & -2 \\ 1 & -2 & 1 \\ \hline\end{array}$ E_8 : $(1,-2,1)^t(1,-2,1)$

Table A.3: Filtros de Newton bidimensionales expandidos.

$\begin{array}{|ccc|}\hline 1 & 1 & -2 \\ 2 & 2 & -4 \\ 1 & 1 & -2 \\ \hline\end{array}$	$\begin{array}{|ccc|}\hline -2 & 1 & 1 \\ -4 & 2 & 2 \\ -2 & 1 & 1 \\ \hline\end{array}$
$M_0 : 0$	$M_4 : \pi$
$\begin{array}{|ccc|}\hline 1 & -2 & -4 \\ 1 & 2 & -2 \\ 2 & 1 & 1 \\ \hline\end{array}$	$\begin{array}{|ccc|}\hline 1 & 1 & 2 \\ -2 & 2 & 1 \\ -4 & -2 & 1 \\ \hline\end{array}$
$M_1 : \pi/4$	$M_5 : 5\pi/4$
$\begin{array}{|ccc|}\hline -2 & -4 & -2 \\ 1 & 2 & 1 \\ 1 & 2 & 1 \\ \hline\end{array}$	$\begin{array}{|ccc|}\hline 1 & 2 & 1 \\ 1 & 2 & 1 \\ -2 & -4 & -2 \\ \hline\end{array}$
$M_2 : \pi/2$	$M_6 : 3\pi/2$
$\begin{array}{|ccc|}\hline -4 & -2 & 1 \\ -2 & 2 & 1 \\ 1 & 1 & 2 \\ \hline\end{array}$	$\begin{array}{|ccc|}\hline 2 & 1 & 1 \\ 1 & 2 & -2 \\ 1 & -2 & -4 \\ \hline\end{array}$
$M_3 : 3\pi/4$	$M_7 : 7\pi/4$

Table A.4: Filtros de Newton modificados y correspondiente orientación.

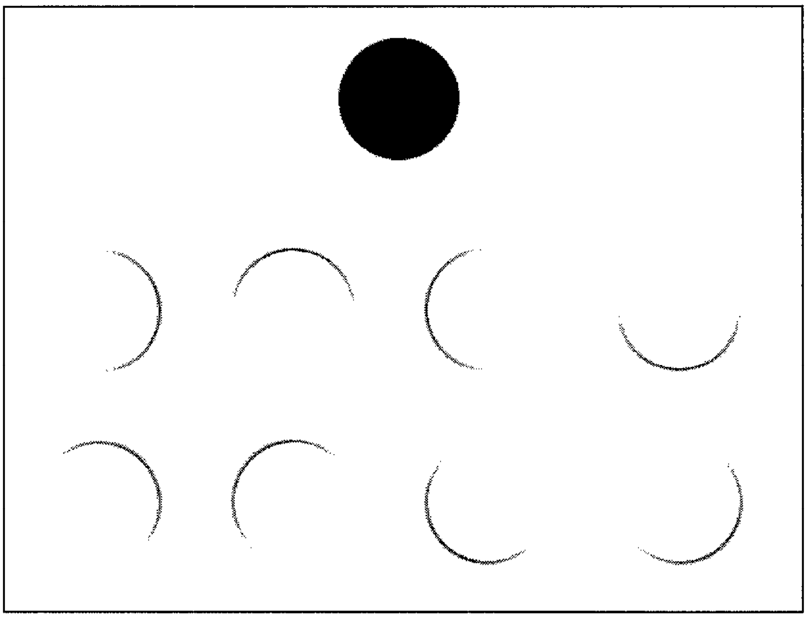


Figure A.5: Salidas positivas de los filtros de Newton modificados para un círculo. Cuanto mayor sea la salida, más oscura ha sido representada.

A.3 Aportaciones originales

A continuación mostramos los puntos más importantes del trabajo, sus aplicaciones y los resultados obtenidos, señalando cuáles son las aportaciones más relevantes. Un conjunto de aplicaciones diferentes han sido propuestas partiendo de un marco y unas unidades básicas comunes:

Estimación de la orientación de los bordes:

Hemos introducido los filtros de Newton modificados para la caracterización de bordes de forma que podemos obtener estimaciones precisas de la orientación que presentan éstos en cualquier punto de una imagen mediante la interpolación de los valores de salida.

Representación y discriminación de formas:

Con las estimaciones de los bordes hemos extraído funciones de orientación que nos permiten describir y caracterizar formas a partir de una representación unidimensional de sus contornos. El uso de funciones de energía a partir de los coeficientes de Fourier de las funciones de orientación, cuyos términos han sido ponderados para realzar su capacidad de discriminación, genera clasificaciones de las formas altamente satisfactorias.

Análisis de movimiento:

Mediante la comparación del contorno de un objeto en diferentes imágenes de una secuencia, hemos extraído los parámetros de traslación, rotación y escalado que proporcionan los elementos necesarios para un análisis de movimiento fiable.

Representación y clasificación multiescala de texturas:

Hemos aplicado técnicas similares a las usadas en la caracterización de formas para la clasificación de texturas mediante el estudio de los contornos en el interior de una región texturada y la construcción de histogramas de orientación que describen la distribución de las orientaciones en esa región permiten. Hemos incorporado el análisis multiescala para generar clasificaciones mucho más robustas, puesto que el estudio de las texturas en diferentes escalas nos aporta una información más completa a la hora de medir la similitud que existe entre ellas, obteniendo con ello un alto grado de discriminación entre texturas.

A.3.1 Estimación de la orientación de los bordes

Los filtros de Newton modificados representan una aproximación nueva en el desarrollo de estructuras en forma de neuronas. En este caso, las salidas iniciales son combinadas de forma que el post-proceso de la información que ellas producen, genera resultados mucho más exactos que aquellos extraídos independientemente. Las propiedades de estos filtros en lo referente a la invarianza rotacional y la no nulidad de los pesos, el proceso de normalización para hacer la salida independiente a los cambios de iluminación, y la interpolación de las salidas discretas generan una extracción de la orientación mucho más refinada.

Con las salidas de los filtros de Newton modificados, analizadas individualmente, podemos discriminar a cuál de las ocho orientaciones principales se parece más un determinado borde. Sin embargo, la distinción entre múltiplos de 45 grados no es suficiente para una descripción y caracterización precisa de los bordes. Por ello, hemos tomado el patrón completo (ver ejemplo en la figura A.6) de los ocho filtros y hemos interpolado entre el valor máximo y sus vecinos mediante un polinomio de orden dos, cuyo máximo viene dado por la siguiente expresión, donde i representa el filtro cuya salida es mayor e $i-1$ e $i+1$ son calculados módulo 8.

$$x_{\max} = \frac{4(f_i - f_{i-1}) - (f_{i+1} - f_{i-1})}{2\left[2(f_i - f_{i-1}) - (f_{i+1} - f_{i-1})\right]}\frac{\pi}{4} + \frac{\pi}{4}(i-1)$$

Con este polinomio podemos estimar de forma mucho más exacta la orientación a la cual corresponde una determinada configuración de valores en la región de 3x3 que estamos analizando.

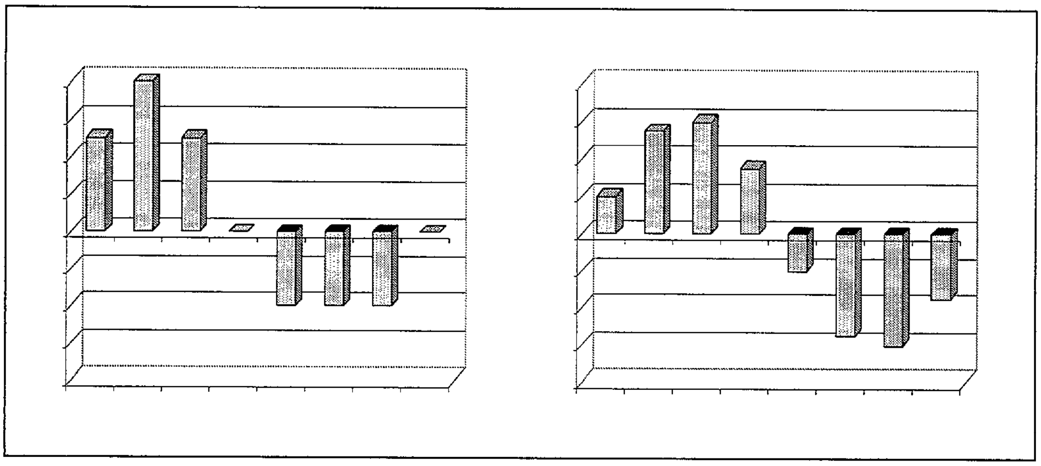


Figure A.6: Salida para una orientación principal (izquierda) y salida para una orientación no principal (derecha).

A.3.2 Representación y discriminación de formas

Una vez estimada la orientación del contorno en cada punto de éste, podemos caracterizar una forma mediante una función de orientación unidimensional, en la que se representan los valores que va tomando la orientación de los bordes a medida que recorremos la silueta del objeto. Ciertas condiciones se fijan sobre este tipo de funciones. Así, la diferencia entre dos valores consecutivos debe ser menor o igual que π radianes para que sean suaves. El contorno debe ser cerrado para recorrerlo en su totalidad, y los puntos deben ser recorridos por vecindad para que el contorno sea contínuo. Un ejemplo se muestra en la figura A.7. Estas funciones serán analizadas mediante sus coeficientes de Fourier, lo que nos permitirá compararlas y caracterizarlas. Al estar trabajando con contornos cerrados, cualquier punto dentro de uno de ellos es válido como comienzo. Sin embargo, el lugar sonde se comienza es relevante, puesto que un cambio en éste transforma la función de orientación obtenida.

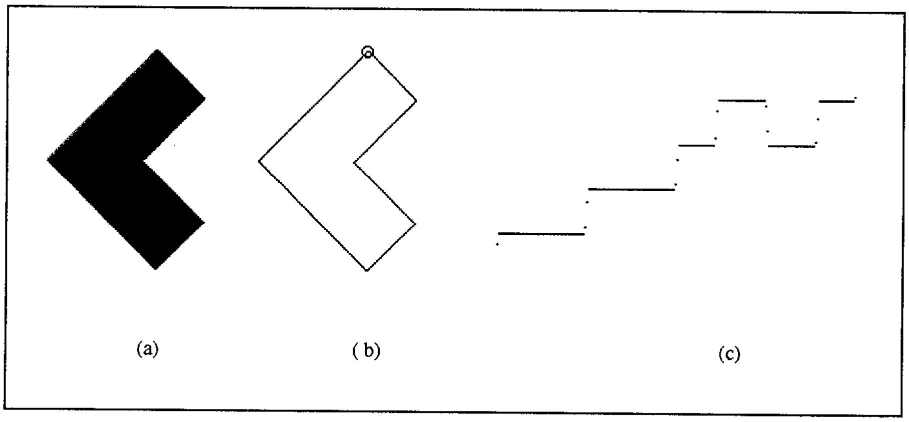


Figure A.7: (a) Imagen original. (b) Contorno y punto de comienzo para la función de orientación. (c) Función de orientación (sentido antihorario).

Al cambiar el punto de comienzo se produce un desplazamiento circular en la función de orientación y ciertos valores de la función, que aparecen en la parte opuesta de ésta, son incrementados en 2π, como se muestra en el ejemplo de la figura A.8. Esto modificará sus coeficientes de Fourier de la siguiente manera, siendo g_n

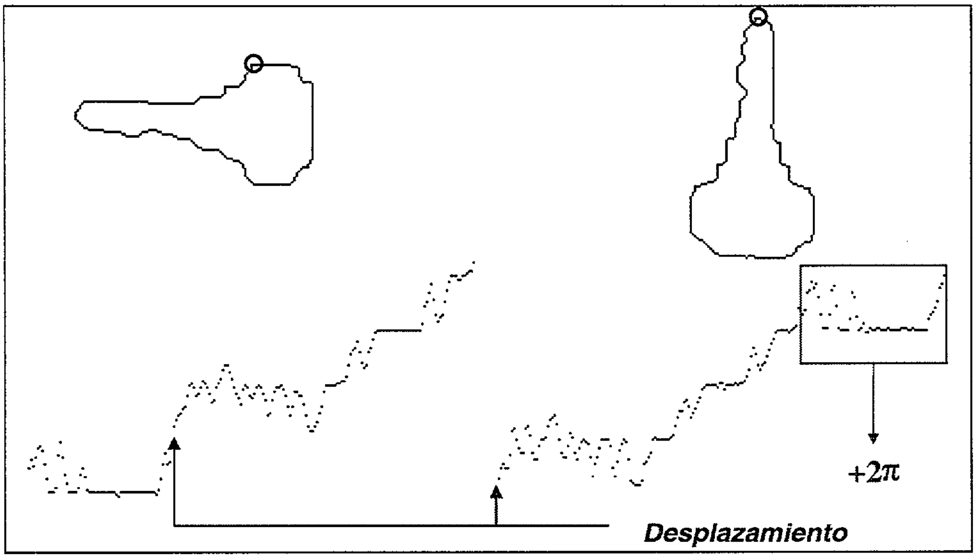


Figure A.8: Efectos de comenzar la función de orientación en un punto diferente.

una versión desplazada a posiciones y corregida de f_n:

$$
\begin{aligned}
\widehat{g}_k = \widehat{f}^a_n &= \frac{1}{L}\sum_{n=0}^{L-1} f^a_n e^{-i\frac{2\pi kn}{L}} \\
&= \frac{1}{L}\left(\sum_{n=0}^{L-a-1} f_{n+a} e^{-i\frac{2\pi kn}{L}} + \sum_{n=L-a}^{L-1} \left(2\pi + f_{n-(L-a)}\right) e^{-i\frac{2\pi kn}{L}}\right) \\
&= \frac{1}{L}\left(\sum_{m=a}^{L-1} f_m e^{-i\frac{2\pi k(m-a)}{L}} + \sum_{m=0}^{a-1} f_m e^{-i\frac{2\pi k(m-a+L)}{L}} + 2\pi \sum_{n=L-a}^{L-1} e^{-i\frac{2\pi kn}{L}}\right) \\
&= e^{i\frac{2\pi ka}{L}} \widehat{f}_k + \frac{2\pi\left(e^{-i\frac{2\pi k(L-a)}{L}} - e^{-i2\pi k}\right)}{L\left(1 - e^{-i\frac{2\pi k}{L}}\right)} \\
&= e^{i\frac{2\pi ka}{L}} \widehat{f}_k + \frac{2\pi\left(e^{i\frac{2\pi ka}{L}} - 1\right)}{L\left(1 - e^{-i\frac{2\pi k}{L}}\right)} \qquad \forall\, k \neq 0
\end{aligned}
$$

A partir de esta relación, podemos estimar cuál es el desplazamiento efectuado para poder comparar las funciones de orientación apropiadamente.

$$
e^{i\frac{2\pi ak}{L}} = \frac{2\pi + L\widehat{g}_k\left(1 - e^{-i\frac{2\pi k}{L}}\right)}{2\pi + L\widehat{f}_k\left(1 - e^{-i\frac{2\pi k}{L}}\right)}
$$

$$a = -\frac{iL}{2\pi k} \ln \left(\frac{2\pi + L\widehat{g}_k \left(1 - e^{-i\frac{2\pi k}{L}}\right)}{2\pi + L\widehat{f}_k \left(1 - e^{-i\frac{2\pi k}{L}}\right)} \right)$$

Sin embargo, esta expresión calcula el desplazamiento sufrido por la señal a partir de una pareja de coeficientes de Fourier. Dado que podemos obtener un valor más exacto considerándolos todos, hemos introducido funciones de energía para medir la similitud de las formas. En ellas determinamos cuán parecidas son dos formas mediante un sumatorio de términos que relacionan los coeficientes de Fourier de las señales de orientación.

$$\widehat{f}_k = \tfrac{1}{L} \textstyle\sum_{n=0}^{L-1} f_n e^{-i\frac{2\pi k n}{L}} \qquad \forall\ k = 0, 1, 2, ..., L-1$$

$$\begin{aligned} E(a) &= \sum_{k=1}^{\frac{L}{2}} \left(e^{i\frac{2\pi k a}{L}} \widetilde{f}_k - \widetilde{g}_k\right) \left(e^{i\frac{2\pi k a}{L}} \widetilde{f}_k - \widetilde{g}_k\right)^* \\ &= \sum_{k=1}^{\frac{L}{2}} \left(\left|\widetilde{f}_k\right|^2 + |\widetilde{g}_k|^2 - e^{i\frac{2\pi k a}{L}} \widetilde{f}_k \widetilde{g}_k^* - \left(e^{i\frac{2\pi k a}{L}} \widetilde{f}_k \widetilde{g}_k^*\right)^* \right) \end{aligned}$$

$$\text{donde } \ \widetilde{f}_k = \widehat{f}_k + \frac{2\pi}{L\left(1 - e^{-i\frac{2\pi k}{L}}\right)}$$

Así, la estimación de la orientación se traduce en la posibilidad de usar esta información local para caracterizar y discriminar formas que son visualmente muy similares y cuyas diferencias son muy sutiles, incluso para un observador humano.

De forma similar, hemos estudiado qué ocurre cuando la forma es reflejada o cuando se sigue el contorno en la dirección opuesta, obteniendo relaciones semejantes que nos permiten estudiar los diferentes fenómenos que pueden presentarse. En el caso de trabajar con secuencias obtenidas en sentido contrario, se detectaría puesto que, en un caso, diferencia entre el principio y el fin de la señal sería positivo, y en el otro sería negativo. Para el caso de formas reflejadas, los coeficientes serían transformados de la siguiente manera, donde g_n es una versión reflejada de f_n y C es una constante que depende del punto de comienzo y el eje de reflexión:

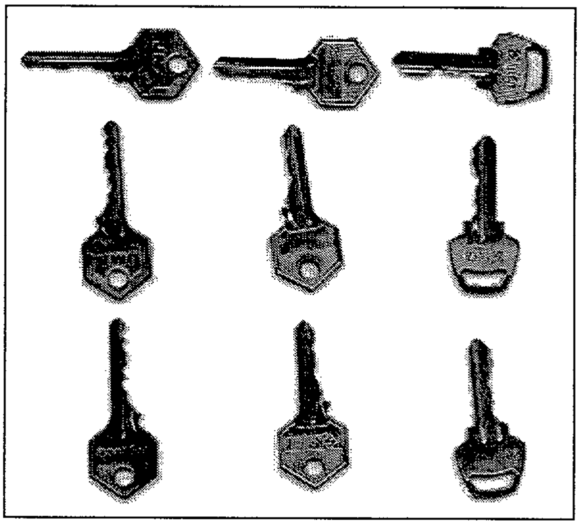


Figure A.9: Imágenes de tres llaves en diferentes posiciones y mostrando ambas caras.

$$
\begin{aligned}
\widehat{g}_k &= \frac{1}{L}\sum_{n=0}^{L-1} g_n e^{-i\frac{2\pi kn}{L}} \\
&= \frac{1}{L}\left(C - f_0 + \sum_{n=1}^{L-1}\left(C - f_{L-n} + 2\pi\right) e^{-i\frac{2\pi kn}{L}}\right) \\
&= \frac{1}{L}\left(C - f_0 + \sum_{n=1}^{L-1}\left(C - f_n + 2\pi\right) e^{i\frac{2\pi kn}{L}}\right) \\
&= \frac{1}{L}\left(-2\pi + \sum_{n=0}^{L-1}\left(C - f_n + 2\pi\right) e^{i\frac{2\pi kn}{L}}\right) \\
&= \begin{cases} -\widehat{f}_0 + \frac{2\pi(L-1)}{L} + C & if \quad k = 0 \\ -\widehat{f}_{-k} - \frac{2\pi}{L} & if \quad k \neq 0 \end{cases}
\end{aligned}
$$

El primer conjunto de prueba está constituido por nueve imágenes de tres llaves diferentes (ver figura A.9). Para poder estudiar los valores, se ha normalizado el conjunto de energías obtenidas dividiendo los valores de la tabla por la media de la energía producida cuando dos imágenes distintas de la misma llave eran comparadas, de forma que los valores cercanos o inferiores a 1 representan formas muy similares, mientras que los que son bastante superiores a 1 representan formas diferentes.

$E_{\min}$	$k_{1:1}$	$k_{1:2}$	$k_{1:3}$	$k_{2:1}$	$k_{2:2}$	$k_{2:3}$	$k_{3:1}$	$k_{3:2}$	$k_{3:3}$
$k_{1:1}$	0.0000	1.1273	0.8219	3.5856	4.1306	4.1161	6.8020	5.9809	5.6455
$k_{1:2}$		0.0000	0.5607	5.4166	5.7934	5.7902	7.5247	6.4634	6.3026
$k_{1:3}$			0.0000	4.5853	5.2063	5.0216	7.4232	6.1860	6.0272
$k_{2:1}$				0.0000	0.9050	1.1134	8.6651	8.5584	8.0708
$k_{2:2}$					0.0000	1.0617	8.9780	6.7763	6.6304
$k_{2:3}$						0.0000	8.8944	8.9579	8.3986
$k_{3:1}$							0.0000	1.3734	1.2102
$k_{3:2}$								0.0000	0.8266
$k_{3:3}$									0.0000

Table A.5: Valores normalizados de la energía mínima.

Estas funciones de energía han sido ponderadas con el fin de realzar la capacidad de caracterización de esta técnica, dando más importancia a aquellos coeficientes que son más significativos para el propósito del trabajo. Más aún, se han testeado varios tipos diferentes de funciones (ver figuras A.10-A.18) para estudiar qué forma debe presentar la función de peso para facilitar la discriminación tanto como sea posible. Dado que cuanto más alto es el orden del coeficiente, más sensible es al ruido, se han tomado funciones decrecientes que conceden una mayor importancia a los primeros coeficientes. El coeficiente de orden cero no ha sido considerado puesto que es el único que se ve alterado por una rotación del objeto y esto provocaría que el resultado no fuese invariante frente a rotaciones.

$$E'(a) = \sum_{k=1}^{\frac{L}{2}} w\left(\frac{2k}{L}\right)\left(\left|\widetilde{f}_k\right|^2 + |\widetilde{g}_k|^2 - e^{i\frac{2\pi ka}{L}}\widetilde{f}_k\widetilde{g}_k^* - \left(e^{i\frac{2\pi ka}{L}}\widetilde{f}_k\widetilde{g}_k^*\right)^*\right)$$

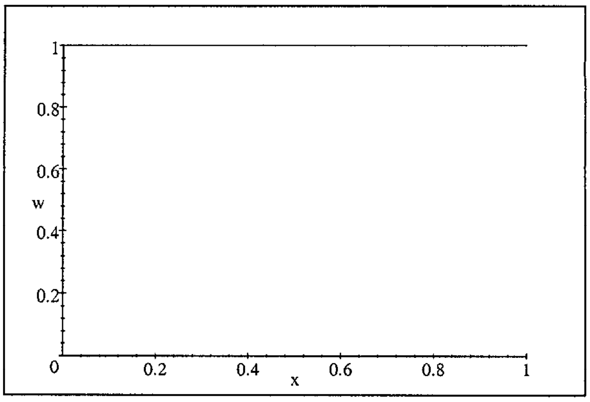


Figure A.10: Función de peso constante 1.

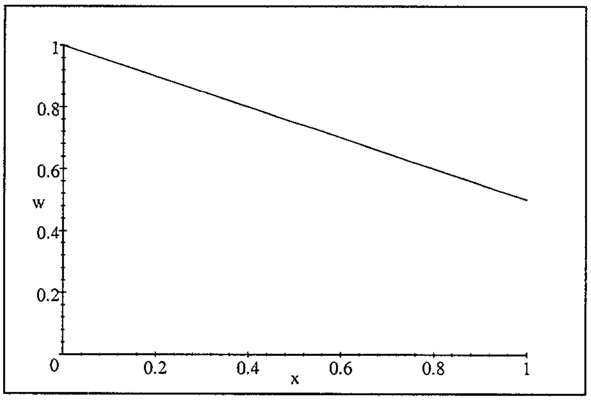


Figure A.11: Función de peso lineal $1 - x/2$.

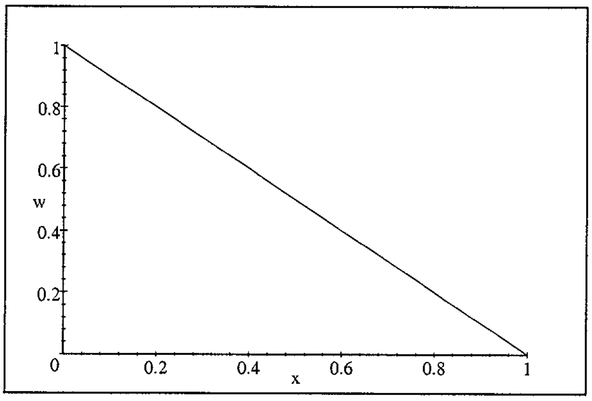


Figure A.12: Función de peso lineal $1 - x$.

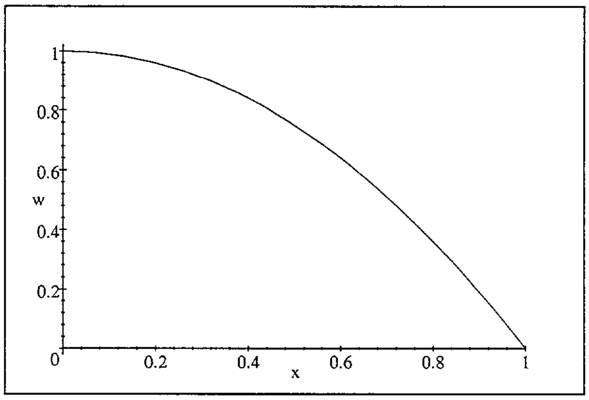


Figure A.13: Función de peso cuadrática $-x^2 + 1$.

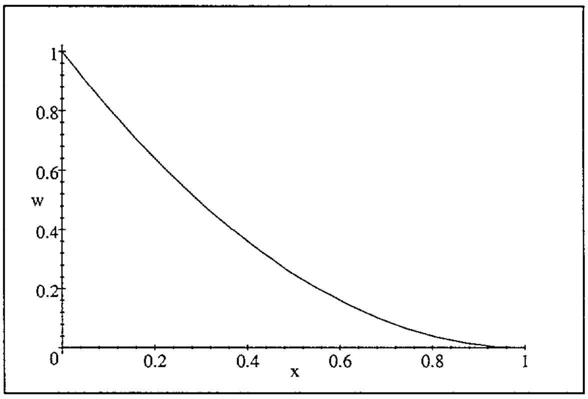


Figure A.14: Función de peso cuadrática $x^2 - 2x + 1$.

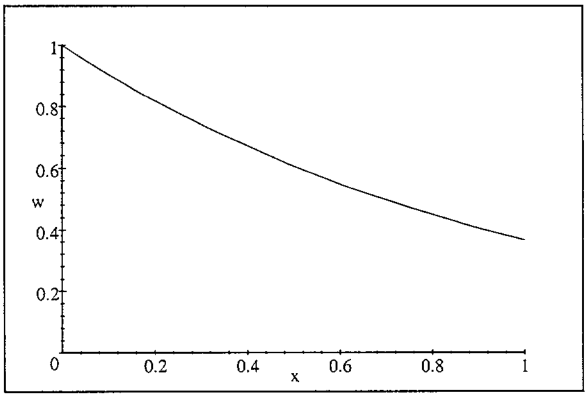


Figure A.15: Función de peso exponencial e^{-sx} para $s = 1$.

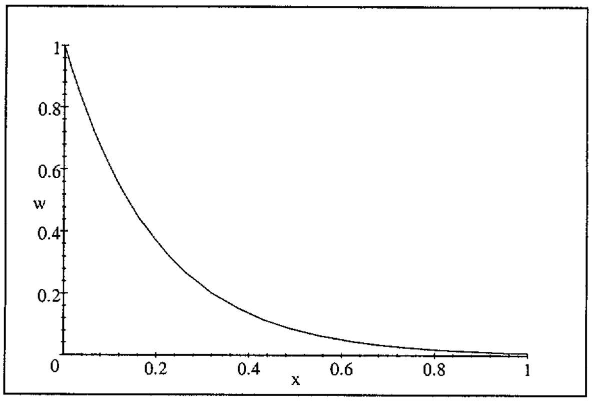


Figure A.16: Función de peso exponencial e^{-sx} para $s = 5$.

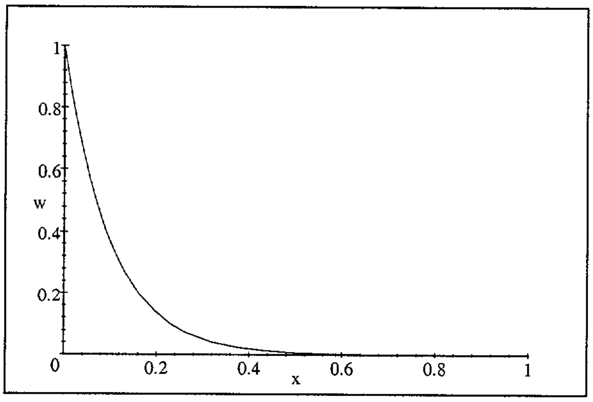


Figure A.17: Función de peso exponencial e^{-sx} para $s = 10$.

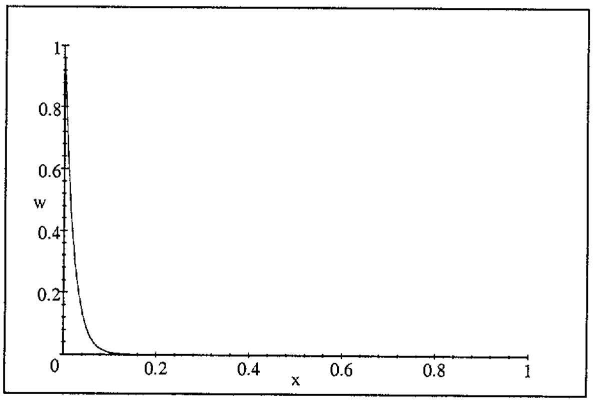


Figure A.18: Función de peso exponencial e^{-sx} para $s = 50$.

Para medir la bondad de una función de peso se ha tomado como parámetro el cociente entre la energía menor entre imágenes de dos llaves diferentes y la energía mayor entre imágenes diferentes de la misma llave. Los resultados obtenidos muestran que las funciones exponenciales generan los mejores valores, y en concreto la función e^{-10x} es la que mejor se comporta. Si comparamos los resultados en la tabla A.5 con los de la tabla A.9, vemos que estos últimos generan una discriminación más clara de las formas.

$w(x)$	$\min(D)$	$\max(S)$	Q_w
1	3.5856	1.3734	2.6107
$1-\frac{x}{2}$	3.7549	1.3957	2.6903
$1-x$	3.9771	1.4250	2.7909

Table A.6: Ratio de discrimninación para funciones constantes y lineales.

$w(x)$	$\min(D)$	$\max(S)$	Q_w
$-x^2+1$	3.7838	1.3972	2.7081
x^2-2x+1	4.2645	1.4663	2.9083

Table A.7: Ratio de discriminación para funciones cuadráticas.

s	$w(x)$	$\min(D)$	$\max(S)$	Q_w
0	1	3.5856	1.3734	2.6107
1	e^{-x}	3.8642	1.4110	2.7386
5	e^{-5x}	4.5941	1.5125	3.0374
10	e^{-10x}	4.8422	1.5318	3.1611
15	e^{-15x}	4.3416	1.4837	2.9662
25	e^{-25x}	3.2275	1.5928	2.0263
50	e^{-50x}	1.7315	1.8574	0.9322
100	e^{-100x}	0.8781	2.0743	0.4233

Table A.8: Ratio de discirminación para funciones exponenciales.

La aplicación de estas técnicas a una base de datos constituida por 1000 imágenes de peces (ver figuras 5.27-5.36 en la versión en inglés) produce una clasificación muy satisfactoria en una situación bastante complicada, como se puede observar en los resultados mostrados en las figuras A.20-A.22.

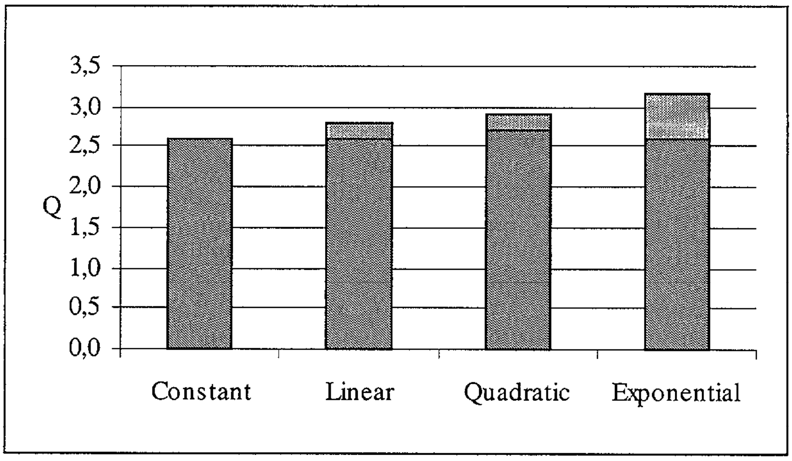


Figure A.19: Resultados obtenidos para funciones constantes, lineales, cuadráticas y exponenciales ($0 \leq s \leq 15$), usadas para ponderar los términos de energía. El gris oscuro y el gris calro señalan el peor y el mejor valor del ratio de discriminación para cada categoría, respectivamente.

$E'_{\min}$	$k_{1:1}$	$k_{1:2}$	$k_{1:3}$	$k_{2:1}$	$k_{2:2}$	$k_{2:3}$	$k_{3:1}$	$k_{3:2}$	$k_{3:3}$
$k_{1:1}$	0.0000	1.1595	0.7754	4.8422	5.4602	4.8694	11.0473	9.0618	8.5987
$k_{1:2}$		0.0000	0.4007	8.2601	8.5063	7.0163	11.9956	9.2841	8.9991
$k_{1:3}$			0.0000	6.7835	6.8951	5.9262	11.8338	9.0418	8.7061
$k_{2:1}$				0.0000	0.8663	1.3553	14.1622	14.3641	13.0408
$k_{2:2}$					0.0000	1.0698	14.5646	12.3600	11.6192
$k_{2:3}$						0.0000	14.3818	15.8590	13.7275
$k_{3:1}$							0.0000	1.5318	1.1737
$k_{3:2}$								0.0000	0.6675
$k_{3:3}$									0.0000

Table A.9: Valores normalizados de nergía mínima con función de peso $w(x) = e^{-10x}$.

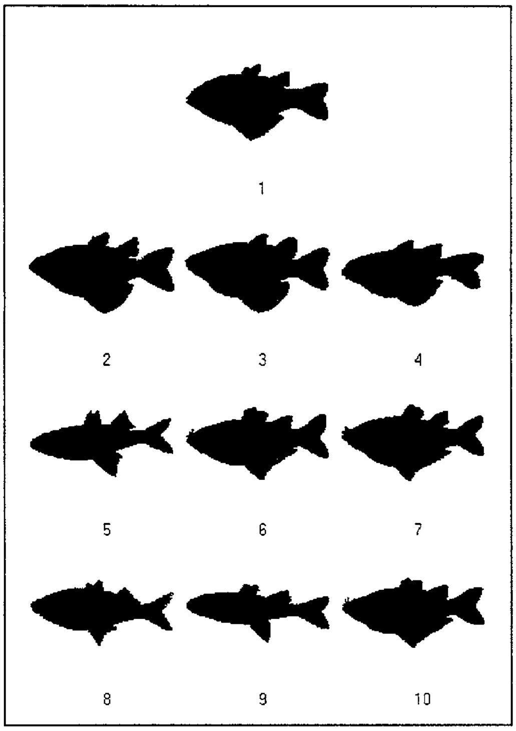


Figure A.20: Ejemplo de los resultados de una búsqueda de formas similares (1).

orden	número de forma	energía ponderada
1	554	0.00
2	553	1279.90
3	552	1535.54
4	569	1997.10
5	521	2424.76
6	557	2613.10
7	560	2617.37
8	487	2625.74
9	529	2795.74
10	558	2855.33

Table A.10: Valores de energía más bajos para la forma 554.

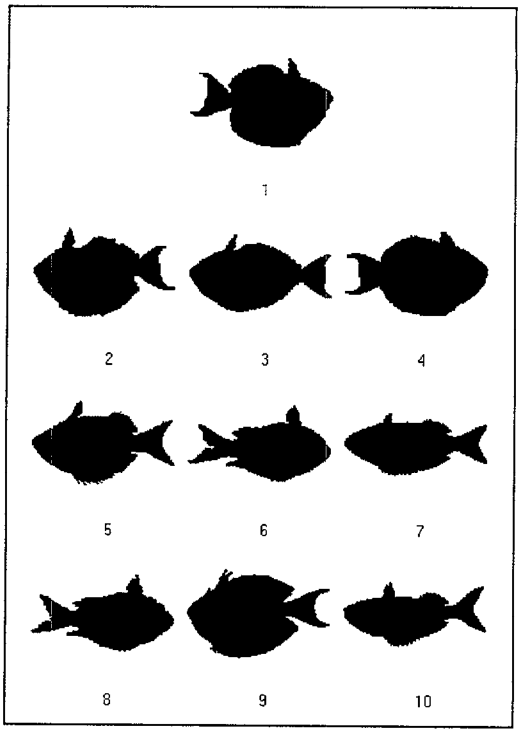


Figure A.21: Ejemplo de los resultados de una búsqueda de formas similares (2).

orden	número de forma	energía ponderada
1	378	0.00
2	372	1818.41
3	373	2213.36
4	381	2438.41
5	863	2619.30
6	244	2839.06
7	851	3332.50
8	222	3394.02
9	371	3509.69
10	856	3602.97

Table A.11: Valores de energía más bajos para la forma 378.

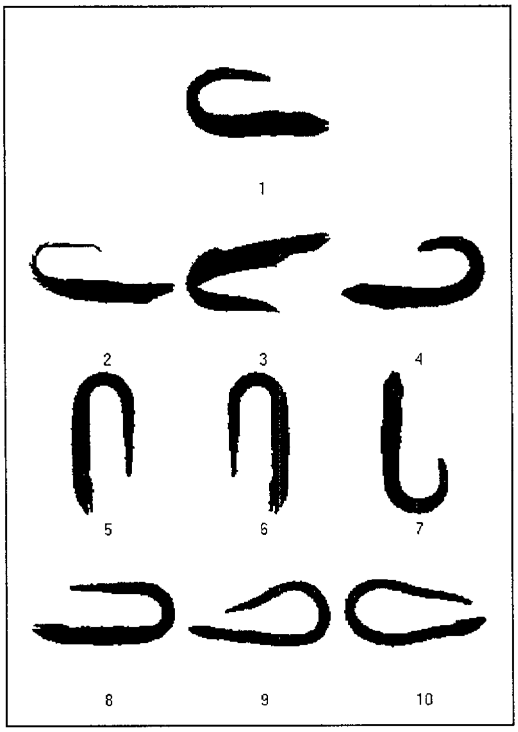


Figure A.22: Ejemplo de los resultados de una búsqueda de formas similares (3).

orden	número de forma	energía ponderada
1	11	0.00
2	385	1508.93
3	570	1522.45
4	67	1549.49
5	79	1910.27
6	183	2094.56
7	186	2453.26
8	181	2516.63
9	103	4354.88
10	83	7700.93

Table A.12: Valores de energía más bajos para la forma 11.

Hemos considerado también la comparación de formas parcialmente ocultas, en las que sólo ciertas partes de las funciones de orientación son emparejables. Para poder asociar segmentos de las funciones de orientación, no sólo deben ser coincidentes entre sí, sino que deben mantener la misma relación de rotación y traslación, de modo que la forma global no sufra otro tipo de transformaciones. Un ejemplo de tal tipo de clasificaciones se muestra en las figuras A.23 y A.24.

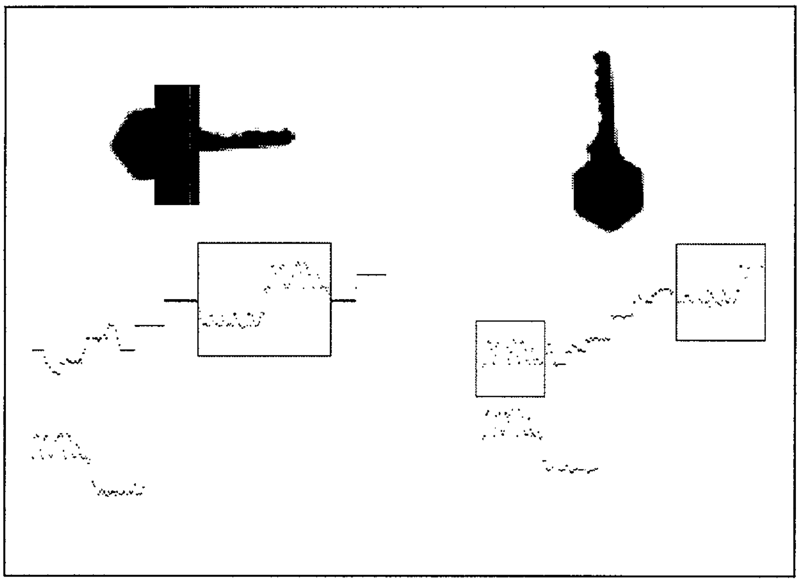


Figure A.23: Parte coincidente de la función de orientación.

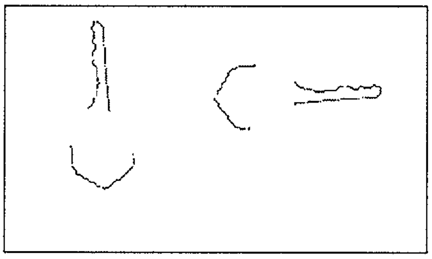


Figure A.24: Asociación de pares de segmentos en el reconocimiento de formas parcialmente ocultas.

A.3.3 Análisis de movimiento

La exactitud de las estimaciones de la orientación de los bordes hace posible extender el rango de aplicaciones a aquéllas en las que el refinamiento de estos valores favorece considerablemente la extracción de parámetros significativos, como el análisis de movimiento. En este caso, proponemos un método para identificar la evolución de un objeto siguiendo una transformación en la que la forma del objeto se mantiene, es decir, incluyendo traslación, rotación y escalado.

Asumiendo esta conservación de la forma, el contorno y los valores de orientación que han sido extraídos para la caracterización de formas hacen posible adaptar una figura a los correspondientes contornos en diferentes instantes de tiempo, y la relación entre estos contornos permite determinar la evolución temporal del objeto.

Las siguientes expresiones muestran cómo se pueden extraer los parámetros de traslación T, escalado S y rotación θ para dos imágenes del mismo objeto, así como la distancia d que mide la exactitud de la estimación. Las figuras A.25 y A.26 muestran un ejemplo de una secuencia y la caracterización del movimiento que se ha extraído.

$$x_c = \frac{\sum_{(x,y)\in I} xO(x,y)}{\sum_{(x,y)\in I} O(x,y)}$$

$$y_c = \frac{\sum_{(x,y)\in I} yO(x,y)}{\sum_{(x,y)\in I} O(x,y)}$$

$$T = (x_c^1 - x_c^2, y_c^1 - y_c^2)$$

$$A = \sum_{(x,y)\in I} O(x,y)$$

$$S = \sqrt{\frac{A^1}{A^2}}$$

$$\theta = \arctan\left(\frac{\sum_{i=1}^{L-1} (x_i^1 y_i^2 - y_i^1 x_i^2)}{\sum_{i=1}^{L-1} (x_i^1 x_i^2 + y_i^1 y_i^2)}\right)$$

$$d(C_1, C_2) = \frac{\sum_{p_1 \in C_1} \min_{p_2 \in C_2} \|p_1 - p_2\|}{|C_1|}$$

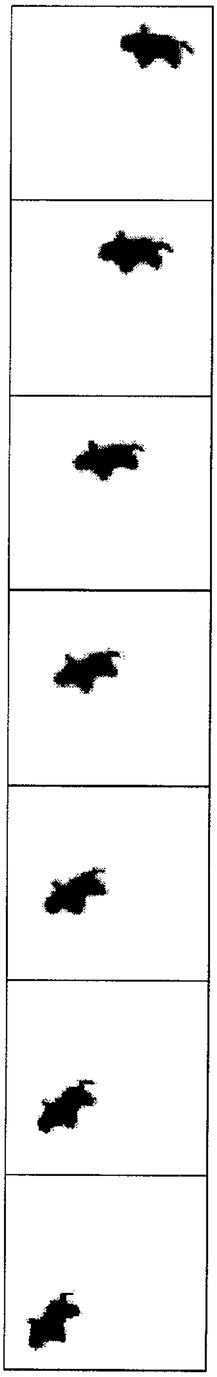


Figure A.25: Secuencia de imágenes de un objeto en movimiento.

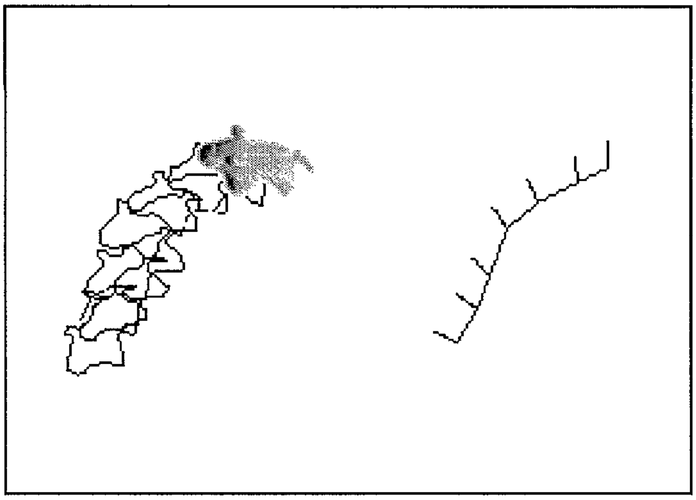


Figure A.26: Secuencia de imágenes y evolución temporal. La trayectoria del objeto se muestra a la derecha y los vectores representados en ella indican los cambios de orientación con respecto a la situación original.

Frames	T_x	T_y	θ	S	d
1-2	-12.6985	-5.5378	0.1332	1.0130	0.4069
2-3	-15.8720	-8.1660	0.2356	0.9602	0.5150
3-4	-12.2985	-10.3341	0.2360	0.9793	0.5317
4-5	-6.7458	-18.3505	0.1306	1.0091	0.5258
5-6	-4.9547	-13.5380	0.2053	0.9880	0.4464
6-7	-7.7208	-12.7441	0.2154	0.9939	0.4866

Table A.13: Traslación (T_x, T_y), ángulo de rotación θ, proporción de escala S y distancia media d para el ejemplo.

A.3.4 Representación y clasificación multiescala de texturas

Avanzando en el estudio de las propiedades de un objeto, hemos aplicado estos filtros a la clasificación de texturas. Para ello hemos estimado la orientación de los contornos en todos los puntos de la región texturada y, a partir de estas estimaciones, hemos contruido un histograma de orientación para cada textura, de forma que en él queda reflejada la distribución de las diferentes orientaciones. La comparación de estos histogramas mediante sus coeficientes de Fourier permite clasificar y relacionar diferentes texturas, utilizando energías similares a las empleadas en la caracterización de formas.

$$E(a) = \sum_{k=1}^{\frac{L}{2}} w\left(\frac{2k}{L}\right)\left(f_k - g_k e^{-i\frac{2\pi k a}{L}}\right)\left(f_k - g_k e^{-i\frac{2\pi k a}{L}}\right)^*$$

Debido al hecho de que las texturas pueden presentar diferentes escalas, y esto podría afectar a los histogramas de orientación que son utilizados para su caracterización, hemos incluido un análisis multiescala para ser capaces de procesar las texturas independientemente de la escala a la que han sido adquiridas. La combinación de los filtros de Newton modificados y el análisis multiescala representa una nueva aproximación en este tipo de procesos que genera clasificaciones apropiadas de las bases de datos de texturas. El análisis multiescala ha sido utilizado primeramente para estudiar la evolución de la suma de los cuadrados de los gradientes dentro de cada textura. Esta evolución nos permite extraer un parámetro de ajuste k que relaciona las escalas de dos texturas de la siguiente manera:

$$k = \frac{\sum_{i=0}^{N-1}\left(r_i^1 r_i^2\right)}{\sum_{i=0}^{N-1}\left(r_i^2\right)^2}$$

donde r_i^1 y r_i^2 son los ratios, para ambas texturas, de la suma de los cuadrados de los gradientes en la escala $i-$ésima con respecto a esa misma suma en la escala inicial. Tras ajustar ambas funciones, podemos considerar el error absoluto entre una de ellas y la otra corregida, de forma que cuanto menor sea el error, más similares son las texturas.

Al igual que con la caracterización de formas, hemos usado una base de datos de texturas para aplicar nuestra técnica (ver figuras 7.12 y 7.13 en la versión en inglés), y a continuación mostramos algunos resultados.

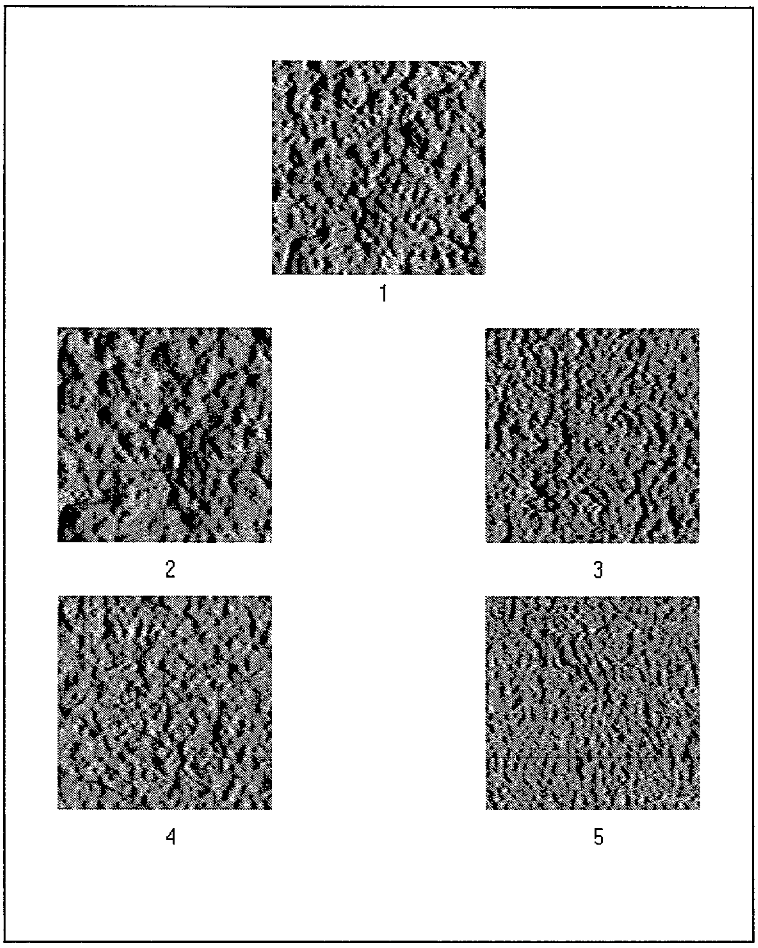


Figure A.27: Ejemplo de los resultados de la búsqueda de texturas similares a la textura 11.

orden	número de textura	energía ponderada
1	11	0.00
2	30	0.59
3	49	0.61
4	10	1.20
5	26	1.62

Table A.14: Valores de energía más bajos para la textura 11.

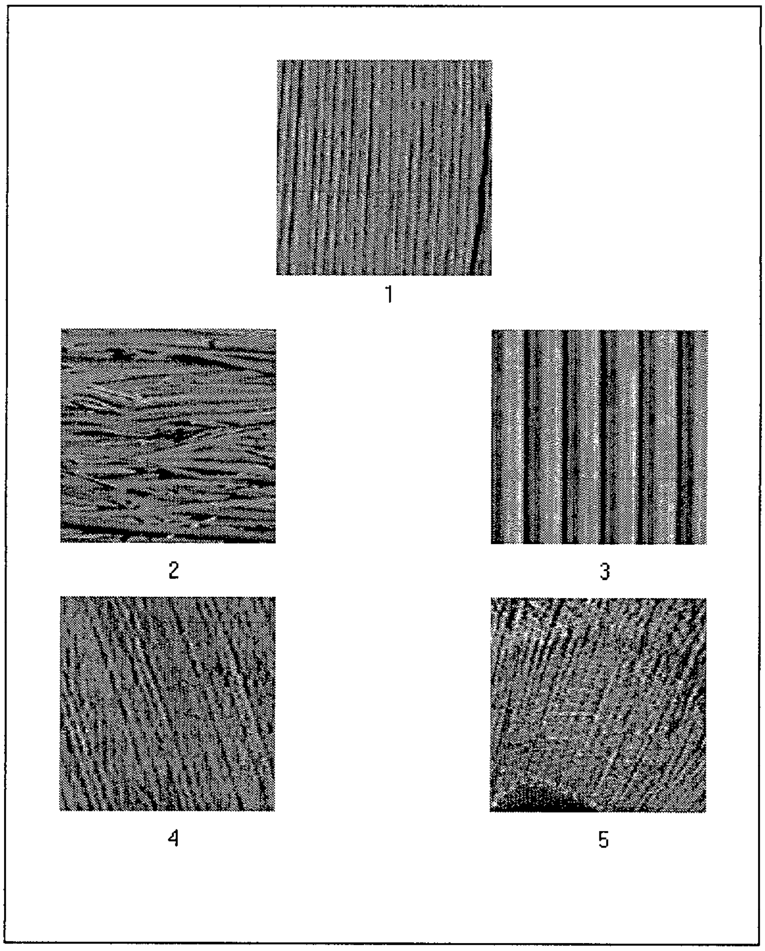


Figure A.28: Ejemplo de los resultados de la búsqueda de texturas similares a la textura 51.

orden	número de textura	energía ponderada
1	51	0.00
2	40	16.03
3	38	49.21
4	56	118.39
5	57	157.10

Table A.15: Valores de energía más bajos para la textura 51.

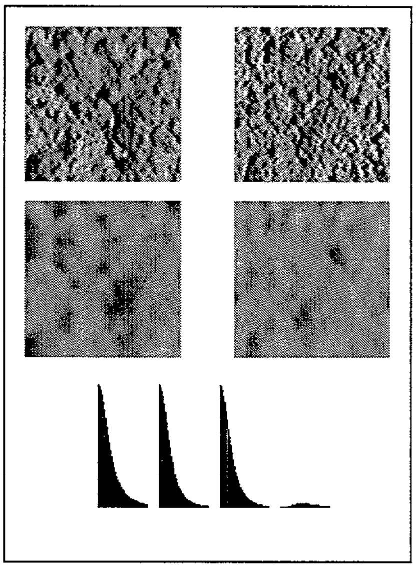


Figure A.29: Texturas 30 y 11.Texturas 30 y 11 después de un filtrado gaussiano. Ajuste de la escala de las texturas y error.

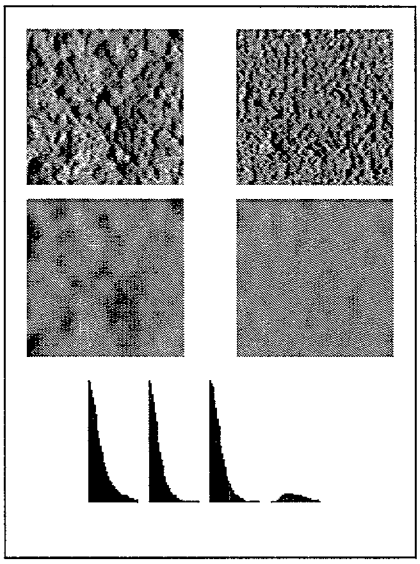


Figure A.30: Texturas 30 y 10.Texturas 30 y 10 después de un filtrado gaussiano. Ajuste de la escala de las texturas y error.

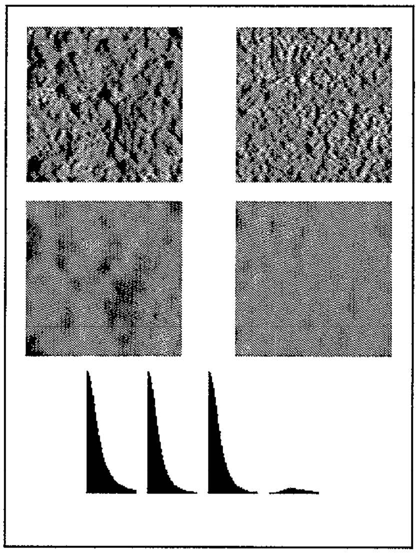


Figure A.31: Texturas 30 y 49.Texturas 30 y 49 después de un filtrado gaussiano. Ajuste de la escala de las texturas y error.

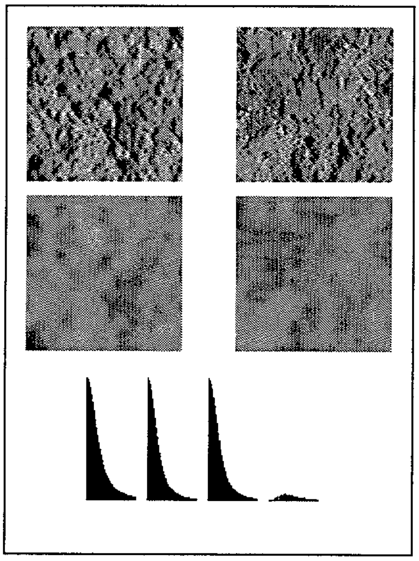


Figure A.32: Texturas 30 y 50.Texturas 30 y 50 después de un filtrado gaussiano. Ajuste de la escala de las texturas y error.

comparación	energía	energía ponderada	error cuadrático (*100)	error absoluto
30-10	3.50	0.47	3.3053	1.035815
30-49	3.66	0.57	10.2568	1.827342
30-11	4.18	0.59	2.2914	0.877211
30-50	3.85	1.07	3.9379	1.111229

Table A.16: Comparación de la tetxura 30 con las más similares.

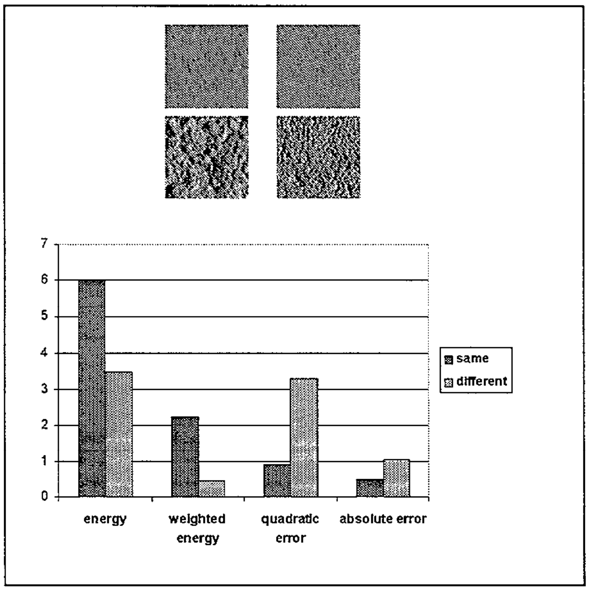


Figure A.33: Comparación entre la energía, la energía ponderada, el error cuadrático y el error absoluto cuando se comparan dos imágenes de la misma textura a diferentes escalas (primera pareja) y dos imágenes de texturas diferentes pero similares (segunda pareja).

Podemos estudiar cómo evoluciona la energía obtenida cuando se comparan los histogramas de orientación de dos texturas I e I' a medida que se van aplicando filtrados gaussianos sobre ellas. Primero calculamos la escala en la que la suma de los cuadrados de los gradientes es la mitad que esta misma suma en la escala inicial, para cada una de ellas. Con estos valores, σ y σ', y el factor de ajuste k, calculamos un valor de escala intermedio σ_N como:

$$\sigma_N = \frac{\sigma + \frac{1}{k}\sigma'}{2}$$

Y finalmente comparamos las energías obtenidas a lo largo de una serie de escalas que progresan como se muestra a continuación:

$$\sigma_n = \frac{n}{N}\sigma_N$$

$$\sigma'_n = \frac{nk}{N}\sigma_N$$

Como se observa en las figuras A.34 y A.35, la comparación de las evoluciones de las energías complementa la información proporcionada por la energía inicial, de forma que la clasificación y las comparaciones son más exactas.

También hemos considerado los efectos del oscurecimiento, aclarado e inversión de las texturas en la energía de los histogramas. Los resultados muestran una gran robustez frente a estos tipos de transformaciones, y en los casos en los que se presentan algunas dudas a la hora de clasificar las texturas, el análisis multiescala aumenta la información, de manera que la caracterización es mucho más precisa.

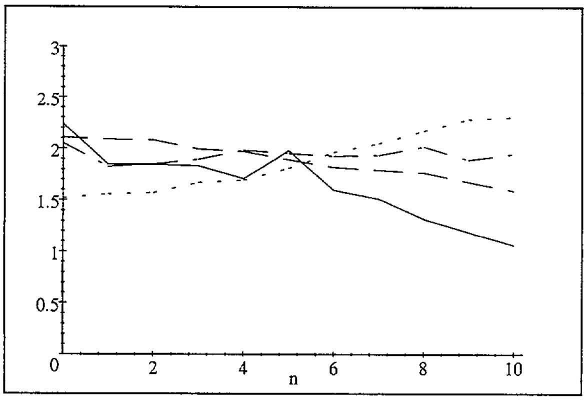


Figure A.34: Evolución de la energía cuando se aplica un filtrado gaussiano al comparar la tetxura 31 con la 26 (puntos), 27 (punto-raya), 4 (rayas) y 12 (sólida).

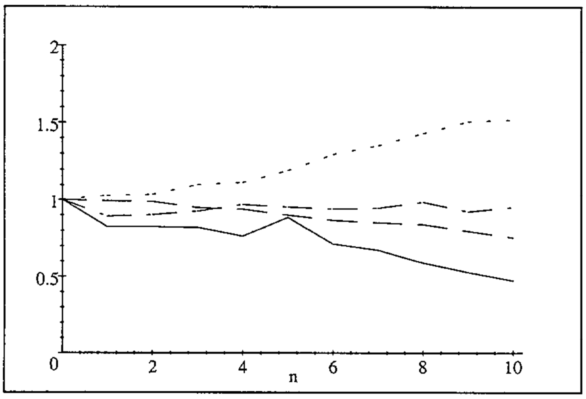


Figure A.35: Evolución relativa de la energía cuando se aplica un filtrado gaussiano al comparar la tetxura 31 con la 26 (puntos), 27 (punto-raya), 4 (rayas) y 12 (sólida).

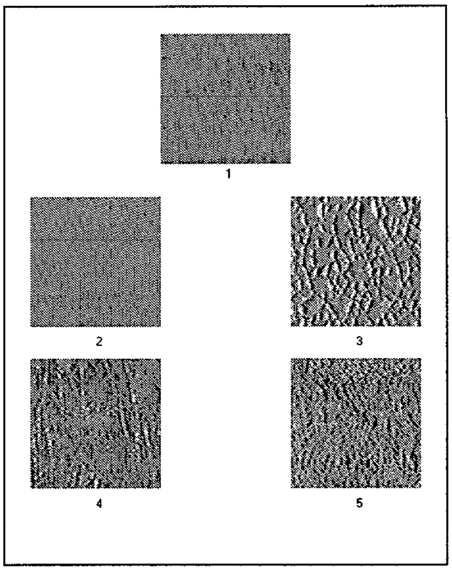


Figure A.36: Resultados ordenados al comparar la textura 31 con las texturas 12, 4, 27 y 26 usando la energía de los histogramas a diferentes escalas con filtrado gaussiano.

comparación	energía inicial	energía final	% energía final-inicial
31-12	2.24	1.06	47.42
31-04	2.11	1.59	75.53
31-27	2.05	1.95	95.07
31-26	1.52	2.31	151.96

Table A.17: Comparación de la tetxura 31 con las más similares usando las energías ponderadas a diferentes escalas.

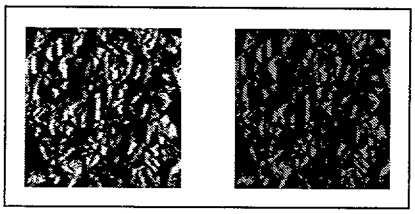


Figure A.37: Textura 15 antes y después del oscurecimiento.

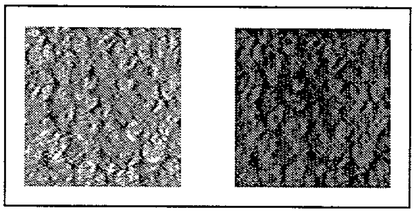


Figure A.38: Textura 18 antes y después del oscurecimiento.

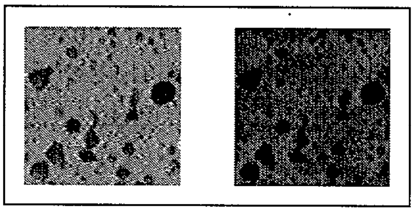


Figure A.39: Textura 21 antes y después del oscurecimiento.

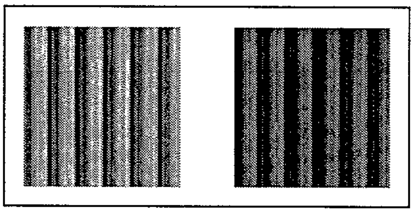


Figure A.40: Textura 38 antes y después del oscurecimiento.

comparación	enegía ponderada
15- oscura 15	0.0083
18- oscura 18	0.0050
21- oscura 21	0.0129
38- oscura 38	0.0145

Table A.18: Comparación con texturas oscurecidas.

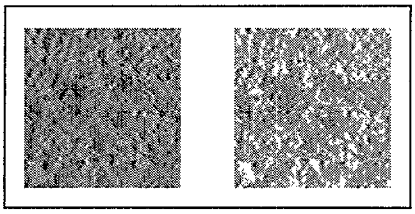


Figure A.41: Textura 3 antes y después del aclarado.

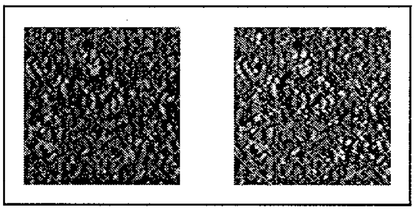


Figure A.42: Textura 14 antes y después del aclarado.

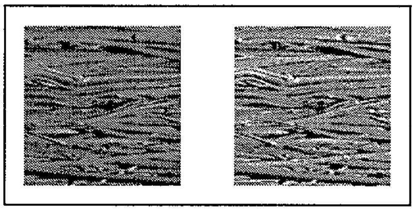


Figure A.43: Textura 40 antes y después del aclarado.

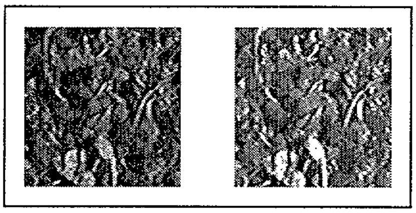


Figure A.44: Textura 53 antes y después del aclarado.

comparación	energía ponderada
03- clara 03	0.0374
14- clara 14	0.0117
40- clara 40	0.0052
53- clara 53	0.0757

Table A.19: Comparación con texturas aclaradas.

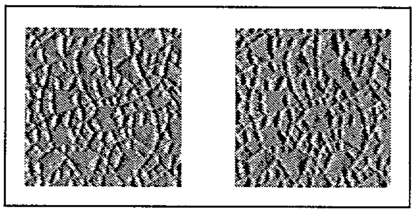


Figure A.45: Textura 4 antes y después de la inversión.

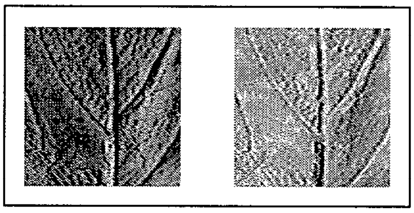


Figure A.46: Textura 23 antes y después de la inversión.

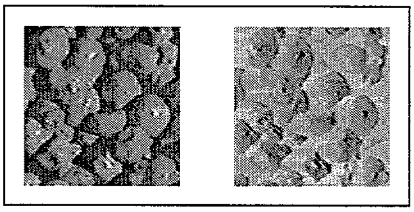


Figure A.47: Textura 35 antes y después de la inversión.

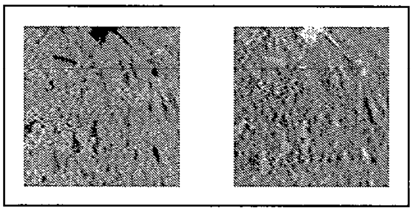


Figure A.48: Textura 52 antes y después de la inversión.

comparación	energía ponderada
04- invertida 04	0.2567
23- invertida 23	0.4859
35- invertida 35	0.6009
52- invertida 52	0.4927

Table A.20: Comparación con texturas invertidas.

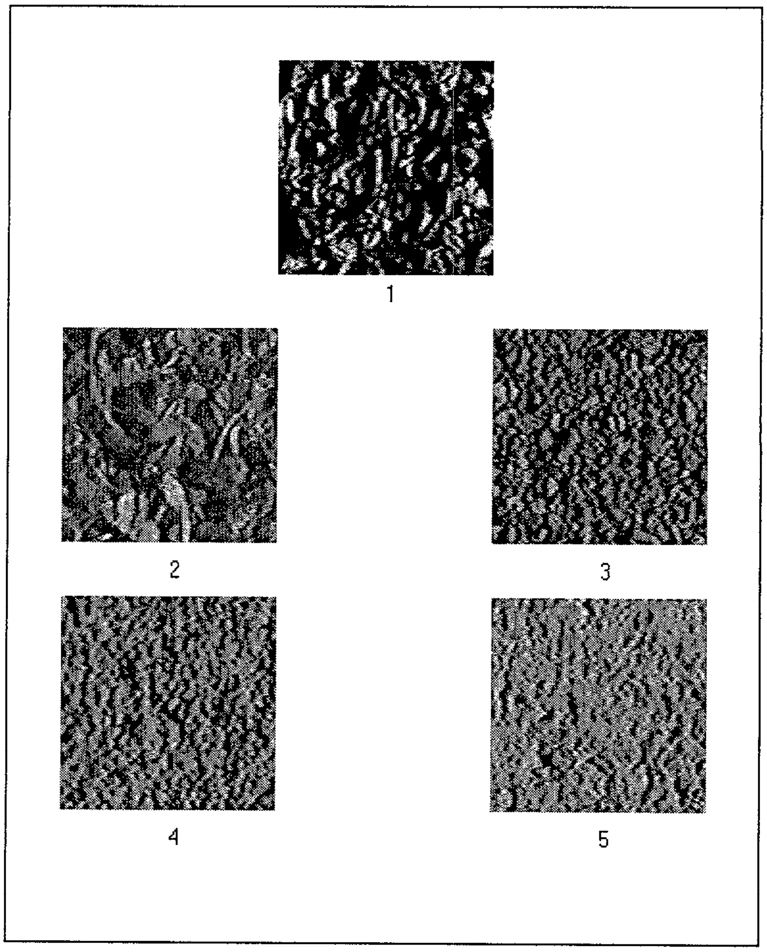


Figure A.49: Resultados de la búsqueda de texturas similares a la textura 15.

orden	número de textura	energía
1	15	0.00
2	53	1.08
3	32	1.43
4	08	2.06
5	49	2.13

Table A.21: Comparación de la textura 15 con las más similares.

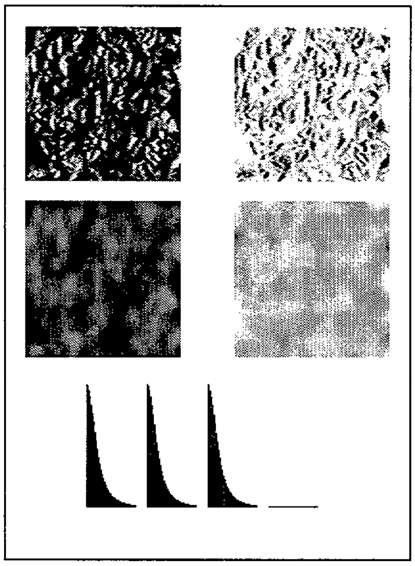


Figure A.50: Textura 15 y su invertida. Textura 15 y su invertida después de un filtrado gaussiano. Ajuste de la escala de las texturas y error.

comparación	energía	energía ponderada	error cuadrático(*100)	error absoluto
15- inverted 15	11.8061	3.0951	0.1007	0.170245

Table A.22: Comparación de la tetxura 15 con su inversa.

comparación	error cuadrático(*100)	error absoluto
15-53	0.0748	0.140708
15-32	2.0305	0.807616
15-08	1.9196	0.766233
15-49	2.4117	0.873240

Table A.23: Comparación de la textura 15 con las más similares.

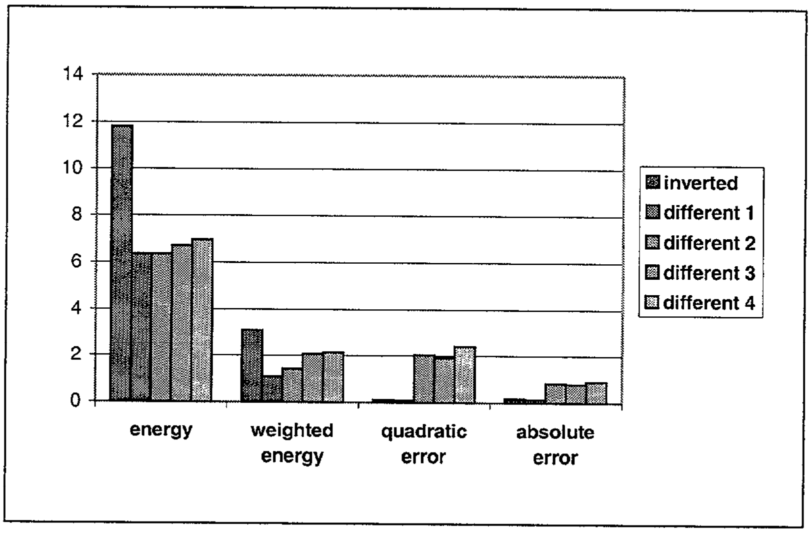


Figure A.51: Comparación entre la energía, la energía ponderada, el error cuadrático y el error absoluto al comparar una textura con su inversa y la misma textura con cuatro texturas similares.

Finalmente, también hemos considerado el papel del color en todos estos procesos, ya que las bases que se han establecido aquí pueden ser ampliamente complementadas tomando otros factores como el color, las imágenes tridimensionales, etc. A partir de la inspiración biológica, hemos construido un sistema que es capaz de procesar la información visual de una escena en diferentes canales. El paralelismo de muchas de las tareas que han sido implementadas y la interacción entre diferentes subsistemas hacen posible diseñar una estructura efectiva para tal tipo de computaciones. Por otro lado, las unidades básicas son las mismas para varios subsistemas, tales como la representación de formas, el análisis de movimiento y la clasificación de texturas. La aplicabilidad, adaptabilidad y precisión de estas técnicas, junto a su modularidad, las convierten en apropiadas para el desarrollo de sistemas guiados por la visión y la simulación de sistemas naturales.

A.4 Conclusiones y futuras líneas de investigación

El presente trabajo propone un método para la caracterización de formas, el análisis de movimiento y la clasificación de texturas basado en un nuevo conjunto de herramientas para el cálculo y la identificación de los bordes. Éstos nuevos filtros están inspirados en los filtros de Newton y permiten un procesamiento preciso de la información local para construir una representación global de una forma. Debido a las limitaciones y desventajas de los filtros de Newton, algunas modificaciones se han efectuado en ellos con el fin de proveerlos de invarianza rotacional, pero preservando su invarianza frente a cambios globales de iluminación. Otros conjuntos de filtros similares han sido propuestos con anterioridad, como los descritos por Sobel, Kirsch, Robinson y Prewitt, pero nosotros hemos adaptado las características de nuestros filtros a los requerimientos de los procesos que llevamos a cabo.

La invarianza frente a cambios globales de iluminación se alcanza mediante un proceso de normalización. Por otra parte, la invarianza rotacional se obtiene usando pesos de los filtros distribuidos de forma cíclica en la configuración del conjunto de filtros. Esto permite comparar las salidas con un patrón a fin de caracterizar las formas, de modo que identificamos la orientación de los bordes de forma precisa, lo cual es muy importante cuando se estudia la curvatura, los puntos singulares o la deteción selectiva del movimiento de un objeto. Incluso mejoramos las estimaciones mediante la interpolación de los valores del patrón proporcionado por los filtros para encontrar un valor aún más preciso.

A partir de este tipo de filtros básicos es posible construir funciones de orientación que pueden identificar claramente las formas y extraer información global para encajar objetos en patrones previamente descritos. Utilizamos la transformada de Fourier discreta como herramienta fundamental en el análisis. El uso de los coeficientes de Fourier proporciona resultados robustos y permite reducir el coste computacional. Un mecanismo similar para la identificación de formas, basado en series continuas de Fourier, es usado por Zahn y Roskies, pero orientado a formas poligonales con series de puntos irregularmente separados, mientras que en nuestro caso los puntos son equidistantes y usamos la transformada rápida de Fourier como herramienta básica para el análisis de las formas. Más aún, la introducción de una función de peso que afecta a la contribución de cada término en la función de energía permite regular las frecuencias que serán más relevantes en la discriminación.

La forma en que se realizan las computaciones y la simplicidad de las unidades básicas, por medio de las cuales se construyen las más complejas, permiten una implementación paralela y una estructura en capas. Esto hace estos filtros apropiados como modelo de computación retinal. Más aún, la compleción de este conjunto de filtros es un aspecto importante en lo referente a la conservación de la información si consideramos la localización y caracterización de los cambios como principal objetivo.

La información extraída para un segmento de la función de orientación permite relacionar diferentes partes de una secuencia de forma que podamos asociar las partes visibles de un objeto cuando éste está parcialmente oculto. Esto requiere un análisis común de los diferentes segmentos que han sido extraídos, ya que se deben fijar ciertas condiciones para garantizar que ciertamente pertenecen a la misma forma. Así, las relaciones de traslación, rotación y escalado deben ser las mismas para todas la parejas de segmentos asociados.

Más aún, el hecho de que la información proporcionada por estos mecanismos es mayor y más exacta que la obtenida mediante una simple localización de los bordes nos permite usarlos para un fiable análisis de movimiento. En este caso, los contornos que deben asociarse no pertenecen a diferentes escenas, sino a diferentes instantes en un vídeo. Una vez que las formas han sido analizadas y emparejadas, las transformaciones que llevan una de ellas sobre la otra hacen posible extraer parámetros de movimiento, tales como traslación, rotación y escalado, en el tiempo.

Algunas aplicaciones complementarias se extraen directamente de las propiedades de estos filtros. Por ejemplo, el análisis y la clasificación de texturas se implementa fácilmente mediante una comparación de histogramas. En este caso, no es el contorno del objeto lo que se analiza, sino el contorno en cada punto dentro de la región texturada, puesto que es el patrón contenido dentro de la región lo que caracteriza a la textura. Gracias a la estimación aproximada de la orientación del gradiente, podemos determinar la distribución de las diferentes orientaciones dentro de un área, y con técnicas similares a las usadas con las funciones de orientación, podemos establecer una relación entre histogramas de orientación. Sin embargo, la no inyectividad de la generación de estos histogramas, puesto que texturas bastante diferentes pueden generar histogramas de orientación muy similares, hace necesario el uso de otra herramienta que realce la discriminación. La introducción del análisis multiescala proporciona una herramienta potente para este propósito, puesto que, incluso si los histogramas de orientación pudieran ser muy similares para texturas claramente diferentes, su evolución a medida que se les aplica un filtrado gaus-

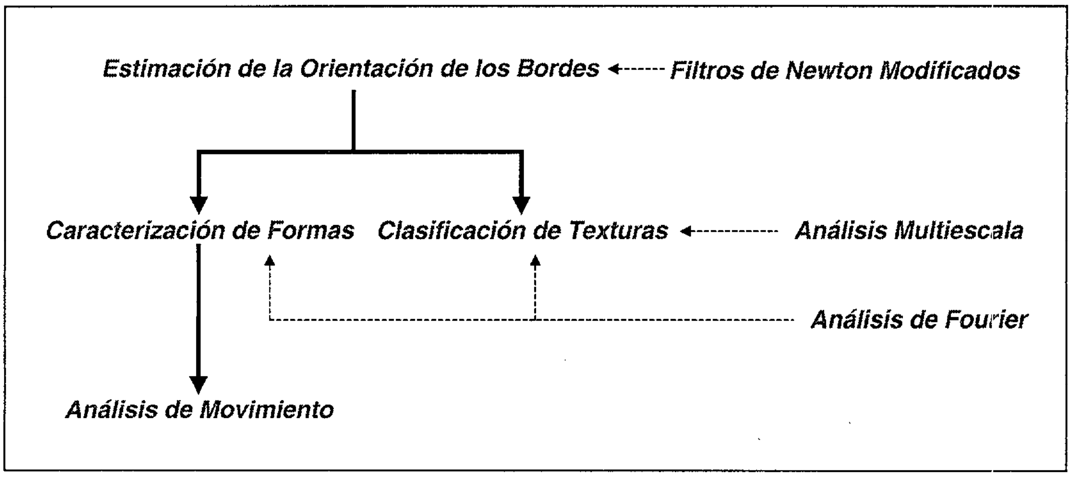

Estimación de la Orientación de los Bordes
Filtros de Newton Modificados
Caracterización de Formas
Clasificación de Texturas
Análisis Multiescala
Análisis de Fourier
Análisis de Movimiento

Figure A.52: Esquema general del trabajo.

siano será diferente, permitiendo así distinguirlos. Esto incluso permite determinar la relación de escalas de dos imágenes obtenidas de la misma textura a diferentes resoluciones.

Las características descritas anteriormente y los resultados obtenidos en las diferentes aplicaciones que de estas técnicas se han realizado muestran la utilidad de estos filtros cuando el objetivo no es meramente la localización de los bordes, sino también la discriminación de su orientación.

En el futuro, sería interesante ampliar estos métodos a sistemas con diferentes canales de color de manera que la información sería triplicada, si usamos un canal de rojo, otro de verde y otro de azul, o analizada de forma individual para la intensidad, la saturación y el brillo. Asimismo, otras ampliaciones significativas podrían tratar la visión binocular y las imágenes tridimensionales. Otros problemas abordables son la segmentación de escenas o la presencia de objetos con múltiples contornos.