Modified Newton Filters for Edge Orientation Estimation, Shape Representation and Motion Analysis

M. Alemán-Flores, L. Álvarez-León and R. Moreno-Díaz jr.
Departamento de Informática y Sistemas
Universidad de Las Palmas de Gran Canaria
Campus de Tafira, 35017 Las Palmas, SPAIN
{maleman, lalvarez, rmorenoj}@dis.ulpgc.es

Abstract

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, it is important that changes in the orientation, size or contrast between the object and the background do not alter the output of the mechanism used for the recognition. In this article, we present a set of formal tools, based on Newton filters, 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 vertebrates retina. The outputs of these filters are used to build a one-dimensional representation of the contour in order to characterize shapes by using Fourier coefficients. They are also used to compare frames of a temporal sequence and extract motion information.

Keywords: Edge Orientation, Shape Representation, Motion Analysis, Newton Filters, Fourier Transform.

1 Introduction

This article presents a new set of tools, the modified Newton filters, and their applications to edge orientation estimation, shape representation and motion analysis. Many sets of filters can be found in the literature to locate the edges of an object. Among them, we can find Newton filters [Mor93]. However, our goal is not the detection itself, but the characterization of edges according to their orientation, since this provides much more

information about the shapes which are presented and allows a further analysis of the objects. From Newton filters, which have been previously used as neuron-like structures for retinal processing [QAM99], a new set is built with some advantageous features. We show how these filters are obtained and how they are used for estimating edge orientation.

The outputs which these 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 these transformations alter the coefficients, we have determined how the contours could be compared and fitted. Finally, the study carried out on the contours of a sequence of images allows extracting some general parameters of an object, such as its position, orientation and size, in order to analyze its temporal evolution.

The article is structured as follows: In section 2, Newton filters are introduced and their mathematical properties are described. Section 3 shows 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. In section 4, 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. Section 5 shows how the selection of the starting point in the input image affects those coefficients. Section 6 explains how we can deal with inversely obtained contour representations and reflected shapes. Section 7 introduces the energy function which is used to measure the similarity between two shapes. Section 8 uses the parameters extracted from the contours to fit objects and analyze their motion. Finally, in section 9, a brief discussion about the previous topics is presented.

2 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 into a layer is the same for each unit inside it, regardless of the position on which it is performed. Nevertheless, the operations of different layers may vary. The mathematical properties of this kind of filters were deeply studied in [Mor93] and we explain some of them below, since they are important for the next sections.

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 figure 1. It is also important to emphasize that, if a filter contains at least one subtractive layer, the sum of all weights is zero.

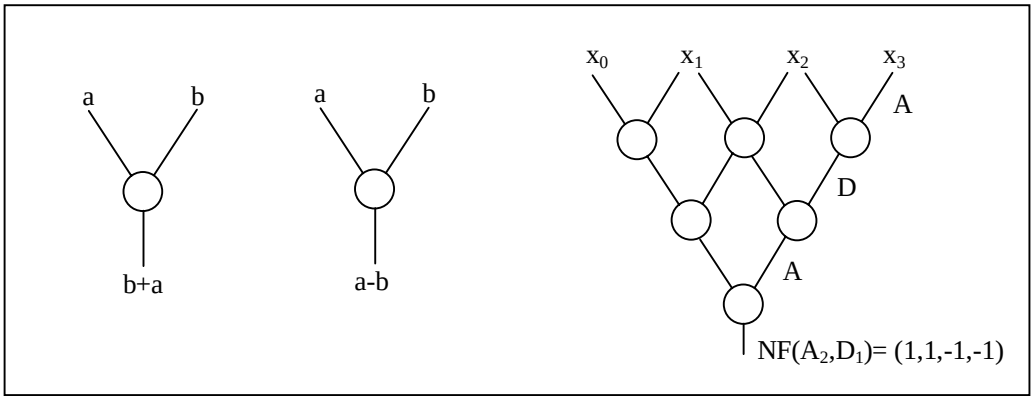

Figure 1: Newton filters structure. Basic operations and example of a filter with two additive layers and one subtractive layer.

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 computing these L filters on them. If we build the corresponding LxL Newton matrix by setting the weights of each filter into a different row, we obtain the following type of transformations:

$$Ax = \begin{pmatrix} a_{0,0} & a_{0,1} & ... & a_{0,L-1} \\ a_{1,0} & a_{1,1} & ... & a_{1,L-1} \\ ... & ... & ... & ... \\ a_{L-1,0} & a_{L-1,1} & ... & a_{L-1,L-1} \end{pmatrix} \begin{pmatrix} x_0 \\ x_1 \\ ... \\ x_{L-1} \end{pmatrix} = \begin{pmatrix} F_0 \\ F_1 \\ ... \\ F_{L-1} \end{pmatrix} = F$$

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. 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

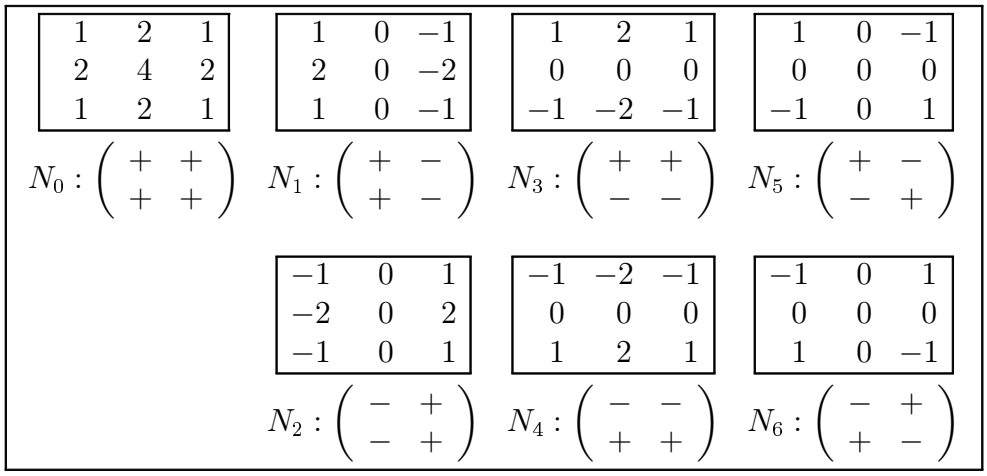

Table 1: Original two-dimensional Newton filters weights.

to recover the original signal very similar to that followed to apply Newton filters [Mor93] [QAM99].

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

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

We can combine one-dimensional filters to build a new set of structures which operate on two-dimensional data. In [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 zero if there are subtractions. The seven filters which are obtained, shown in table 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.

$$\begin{array}{rcl} NF(A_2, D_0) & = & (1, 2, 1) \\ NF(A_1, D_1) & = & (1, 0, -1) \\ NF(A_0, D_2) & = & (1, -2, 1) \end{array}$$

Next, we combine these 3 filters in the horizontal and vertical directions and the set resulting is shown in table 2. According to the function they perform, the first one is a

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

Table 2: Expanded two-dimensional Newton filters.

low-pass filter, and it is the only filter in which the sum of all weights is not zero, 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 a discretization 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 discretizations of the following derivatives of the original input data:

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

Furthermore, this set of filters constitute a complete transformation from the point of view of inversibility, i.e. information is preserved, and the original values could be

recovered from the outputs they provide.

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

Experiments on the activity of ganglion cells of higher vertebrates 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 in 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 below. 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:

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

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 following expression:

$$y = x_0 + x_1(e_1 + e_2 + ... + e_L) + x_2(e_1e_2 + e_1e_3 + ... + e_{L-1}e_L) + ... + x_{L-1}(e_1e_2e_3...e_L)$$

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 objects. In fact, Hubel and Wiesel explain the functioning of cells which react to the presence of bars located with a certain slope, while the output decreases when the orientation is different [HW62][HUB88][Kan90]. On the other hand, Barlow, Hill and Levick describe cells whose outputs are modulated

1	1	−2	−2	−4	−2	−2	1	1	1	2	1
2	2	−4	1	2	1	−4	2	2	1	2	1
1	1	−2	1	2	1	−2	1	1	−2	−4	−2
$F_0 : 0$			$F_2 : \pi/2$			$F_4 : \pi$			$F_6 : 3\pi/2$		
1	−2	−4	−4	−2	1	1	1	2	2	1	1
1	2	−2	−2	2	1	−2	2	1	1	2	−2
2	1	1	1	1	2	−4	−2	1	1	−2	−4
$F_1 : \pi/4$			$F_3 : 3\pi/4$			$F_5 : 5\pi/4$			$F_7 : 7\pi/4$		

Table 3: Modified Newton filters and corresponding orientation.

by motion direction [BHL64]. That is why it is so important, not only to locate edges, but also to determine their orientation in an accurate way. 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 ouput than the others. 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:

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

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 mangnitudes. As a result, it is difficult to compare their outputs properly, causing 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:

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

We try to build filters that react to changes in the 8 main orientations. Consequently, it is necessary to use different signs at both sides of the border. It would be desirable that the weights in the central region of the filter were not zero, 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 zero, we multiply the negative region by 2, thus obtaining the filters shown in table 3. In particular, the filter F_k react to changes in the orientation $k\frac{\pi}{4}$.

We notice that the output of these filters is independent of the particular grey-value magnitude of the image border, i.e. F_k is invariant under a grey-level translation of

the form $I \rightarrow I + C$, 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 (difference) 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 translation grey-level transformation in the following sense: we consider a 3x3 neighborhood of an edge pixel, let X be the vector of the grey-value of the 8 neighbor pixels arranged cyclically, let M be the vector $(m, m, m, m, m, m, m, m)^t$ where m is the grey-value of the central edge pixel, and let A be 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 is always equal to 2). Then we can write the relation

$$AX + 2M = F$$

where $F = (F_0, ..., F_7)^t$ is the modified Newton filters output vector. The inverse of matrix A is given by:

$$A^{-1} = \begin{pmatrix} -\frac{11}{48} & -\frac{1}{24} & \frac{1}{16} & -\frac{5}{24} & \frac{5}{48} & -\frac{5}{24} & \frac{1}{16} & -\frac{1}{24} \\ -\frac{1}{24} & -\frac{11}{48} & -\frac{1}{24} & \frac{1}{16} & -\frac{5}{24} & \frac{5}{48} & -\frac{5}{24} & \frac{1}{16} \\ \frac{1}{16} & -\frac{1}{24} & -\frac{11}{48} & -\frac{1}{24} & \frac{1}{16} & -\frac{5}{24} & \frac{5}{48} & -\frac{5}{24} \\ -\frac{5}{24} & \frac{1}{16} & -\frac{1}{24} & -\frac{11}{48} & -\frac{1}{24} & \frac{1}{16} & -\frac{5}{24} & \frac{5}{48} \\ \frac{5}{48} & -\frac{5}{24} & \frac{1}{16} & -\frac{1}{24} & -\frac{11}{48} & -\frac{1}{24} & \frac{1}{16} & -\frac{5}{24} \\ -\frac{5}{24} & \frac{5}{48} & -\frac{5}{24} & \frac{1}{16} & -\frac{1}{24} & -\frac{11}{48} & -\frac{1}{24} & \frac{1}{16} \\ \frac{1}{16} & -\frac{5}{24} & \frac{5}{48} & -\frac{5}{24} & \frac{1}{16} & -\frac{1}{24} & -\frac{11}{48} & -\frac{1}{24} \\ -\frac{1}{24} & \frac{1}{16} & -\frac{5}{24} & \frac{5}{48} & -\frac{5}{24} & \frac{1}{16} & -\frac{1}{24} & -\frac{11}{48} \end{pmatrix}$$

the condition number of these matrices is:

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

which means that A provides a nice transformation from a numerical point of view. Therefore, we can recover the vector X just taking

$$X = A^{-1}(F - 2M)$$

which means that we can recover the complete image border grey-values from the modified Newton filters output F, and the grey value m of the central pixel.

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 and rotating the image a multiple of $\pi/4$ does not alter the magnitude of the output, but only the indices. Figure 2 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.

When the real orientation does not correspond to one of the 8 main directions, we can estimate it by interpolating the higher value in the output vector F with its two neighbors, which provides a more accurate estimation of the orientation. The interpolating polynomial of the second degree results:

$$\begin{aligned} y &= \frac{8(F_{i+1}-F_{i-1})-16(F_i-F_{i-1})}{\pi^2}\left(x-\frac{\pi}{4}(i-1)\right)^2 \\ &\quad +\frac{8(F_i-F_{i-1})-2(F_{i+1}-F_{i-1})}{\pi}\left(x-\frac{\pi}{4}(i-1)\right)+F_{i-1} \end{aligned}$$

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:

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

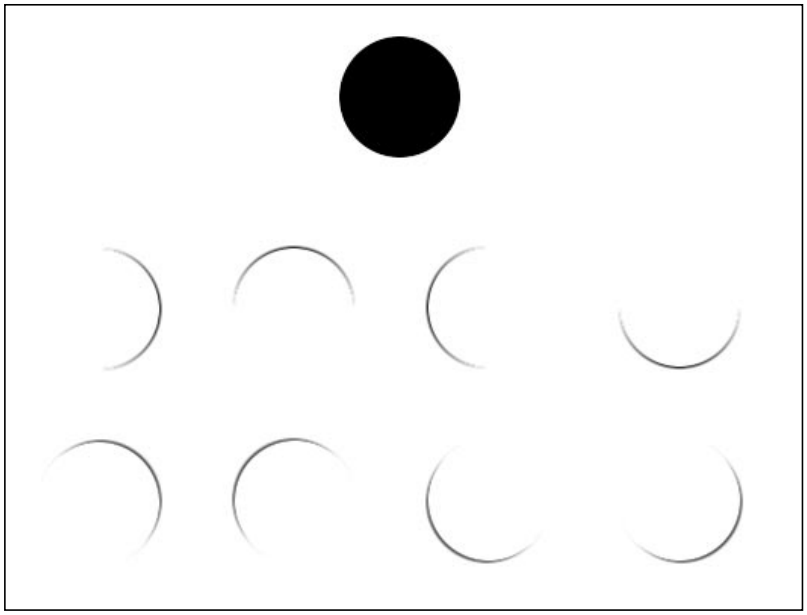

Figure 2: Positive outputs of the modified Newton filters for the circle above (the higher the output value, the darker it has been represented).

For instance, in a non-binary input in which a slope 3 orientation (1.249 radians) is reflected in the values of the pixels according to the proportion of their area covered by the object, as shown in figure 3, 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 $i + 1 = 3$:

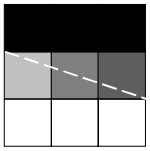

Figure 3: Slope 3 edge.

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

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	$-7/2$

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 as:

$$C(x) = \sum_{i=0}^{7} \frac{P_{(i+x) \bmod 8} F_i}{\|P\| \|F\|} \qquad x \in Z_8$$

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 $x_{\max}$:

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

Similar filters have been proposed by Prewitt, Sobel, Robinson or Kirsch [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, 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 they do not constitute a complete transformation.

The main advantage of this 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, not only the location of borders itself, but also the use of this information for other 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, as done in [ALM97], we can use these filters as basic tools to improve the discrimination.

4 Contour-based shape representation

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

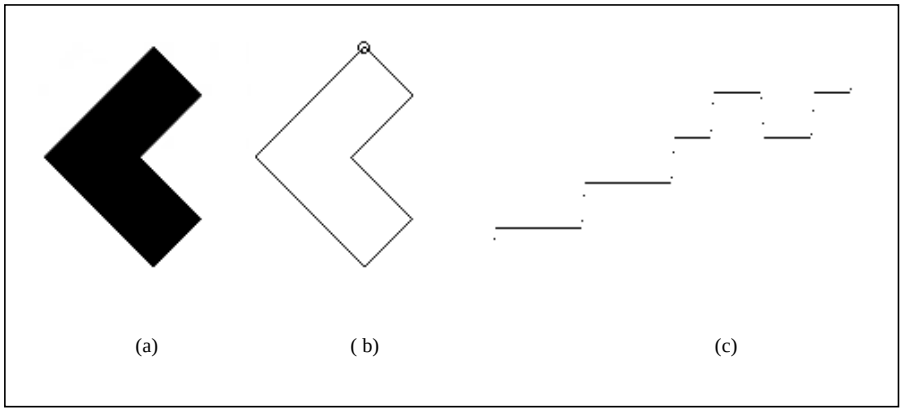

Figure 4: (a) Input image. (b) Contour and starting point for orientation function. (c) Orientation function.

along the border. If the edges are not clear enough, double borders may appear and must be eliminated before continuing.

We must first be placed on a point of the outline and, according to its orientation, find the neighbor that best matches 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.

In figure 4, we can see the orientation of each point belonging to the outline of an object and how this orientation changes. The values of the orientation functions are selected in such a way that no point differs in more than π radians from its neighbors, except for the first and the last points of a closed curve, which are actually neighbors. 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. Different types of one-dimensional shape representations are described in [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. We forced some conditions on this kind of functions, such as selecting values which do not differ in more than π radians from their respective neighbors in order to provide these functions with a certain continuity. 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 at 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. 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 [ZR72], the optimality criteria as well as the consequences of certain transformations on Fourier coefficients are studied in the case of polygonal shapes and using continuous Fourier series. In this paper we generalize the results presented in [ZR72] to the case of uniformly distributed sample set of points of the boundary and we use the Fourier Transform. More recently, in [RSH96], a set of modified Fourier descriptors for shape representation is proposed. 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 the ones presented in [ZR72] and [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:

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

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, not 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 another 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 the following:

$$\widehat{g}_0 = \frac{1}{L}\sum_{n=0}^{L-1} g_n = \frac{1}{L}\sum_{n=0}^{L-1}(f_n + \theta) = \theta + \frac{1}{L}\sum_{n=0}^{L-1} f_n = \widehat{f}_0 + \theta$$

If $k \neq 0$:

$$\widehat{g}_k = \frac{1}{L}\sum_{n=0}^{L-1} g_n e^{-i\frac{2\pi kn}{L}} = \frac{1}{L}\sum_{n=0}^{L-1}(f_n + \theta)e^{-i\frac{2\pi kn}{L}} = \frac{1}{L}\sum_{n=0}^{L-1} f_n e^{-i\frac{2\pi kn}{L}} = \widehat{f}_k$$

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 Influence of the starting point of the orientation sequence

To extract the sequence, any point on the contour can be used as start. However, we must take into account the consequences described in the previous section. 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 at the opposite side of the sequence. Next, we will show some results concerning the influence of the starting point of the orientation sequence for continuous and discrete signals.

5.1 Continuous signals

If $g(.)$ and $f(.)$ are continuous signals, being $g(.)$ a shifted and corrected version of $f(.)$, according to the transformation described above, we can proceed as follows (we use function h_a to add the constant 2π to a part of the signal):

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

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

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

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

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

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

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

$$\text{where } w = 2\pi/L$$

5.2 Discrete 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 in the opposite side in 2π, i.e. they correspond to the same shape starting in a different point of the contour, their coefficients are related as follows:

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

This signal will generate the following coefficients:

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

And consequently, the relationship between the coefficients of both signals is:

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

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

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

Taking into account that order 0 coefficient is related to the mean value of the signal, which 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 follows:

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

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

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 provide 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 section describes how a more accurate value can be obtained.

6 Sequence direction and symmetrical shapes association

We have considered the problem of starting the orientation function of a closed curve in 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 follows:

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

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 following relationship applies:

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

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.

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. 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 the one represented by f_n, starting at the same point, they can be related as follows:

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

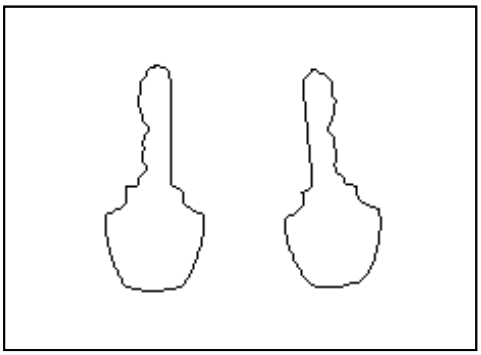

Figure 5: Contours of both sides of the same key.

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. This will modify Fourier transform coefficients as follows:

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

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

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

An example of reflected shape representations is shown in figure 6.

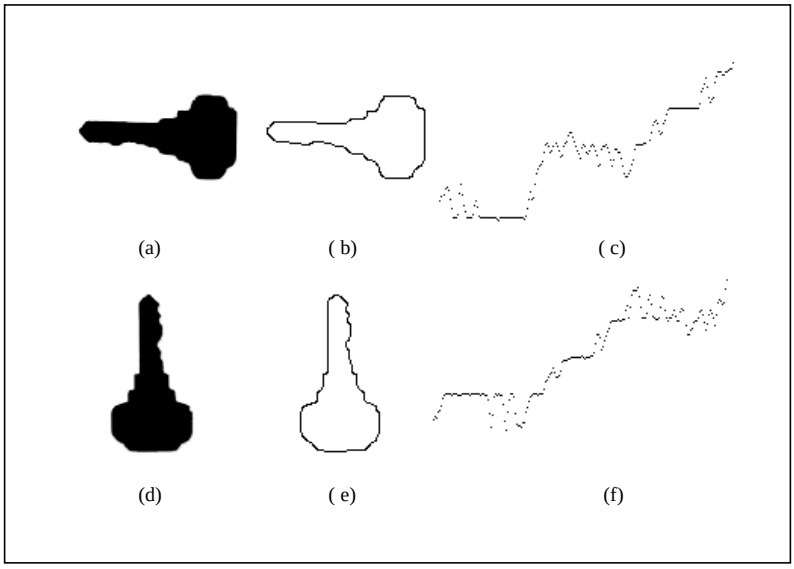

Figure 6: (a) and (d) Input images corresponding to the same key. (b) and (e) Contours. (c) and (f) Orientation functions.

7 Energy function and shape characterization

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 which are not multiples 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 with an addition 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 (1), we can determine how good the relationship is for a certain value of a and a given coefficient order k:

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

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

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

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

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):

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

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

And now, we should extract the value for a which minimizes this function.

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

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

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

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

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

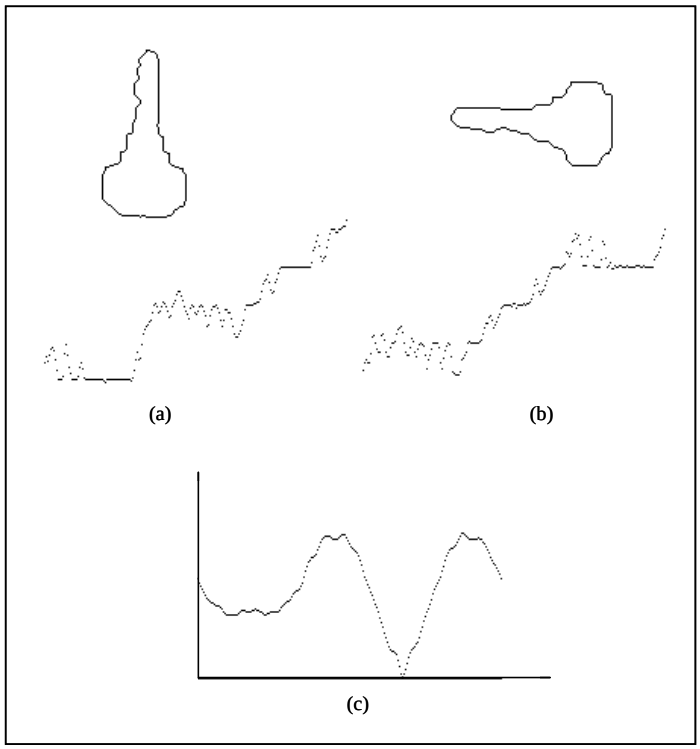

Figure 7: (a) and (b) Contours and orientation functions of two images corresponding to the same key. (c) Energy function $E(a)$ for shift estimation.

Figure 7 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. We use function $M(a)$ as a simplified expression of $dE(a)/da$, whose zero-crossings must be found.

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

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

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

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:

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

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.

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 (2) in such a way that the first coefficients are more significant than the last ones:

$$
W(a) = \sum_{k=1}^{\frac{L}{2}} \frac{L-k+1}{L} \left(\left| \widetilde{f}_k \right|^2 + |\widetilde{g}_k|^2 - e^{i\frac{2\pi k a}{L}} \widetilde{f}_k \widetilde{g}_k^* - \left(e^{i\frac{2\pi k a}{L}} \widetilde{f}_k \widetilde{g}_k^* \right)^* \right) \tag{3}
$$

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 not 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. 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

$E_{\min}$	$k1:1$	$k1:2$	$k1:3$	$k2:1$	$k2:2$	$k2:3$	$k3:1$	$k3:2$	$k3:3$
$k1:1$	0.0000	1.1422	0.8456	3.5392	4.1323	4.1017	6.7139	5.8715	5.5170
$k1:2$		0.0000	0.5715	5.3267	5.7249	5.7384	7.4869	6.4014	6.2206
$k1:3$			0.0000	4.5825	5.2142	5.0398	7.3575	6.0796	5.9305
$k2:1$				0.0000	0.8665	1.1173	8.5641	8.5237	7.9216
$k2:2$					0.0000	1.1437	8.8598	8.7796	6.5387
$k2:3$						0.0000	8.8109	8.6807	8.3070
$k3:1$							0.0000	1.3393	1.1904
$k3:2$								0.0000	0.7832
$k3:3$									0.0000

Table 4: Normalized minimum energy values for keys in figure 8.

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.

Figure 8: Images of three different keys in different positions and showing both sides.

In table 4, the normalized minimum energy values are shown for the comparisons of 9 different images corresponding to 3 keys (see figure 8). In this case, equation (2) has been used and $kn:m$ corresponds to the m^{th} image of the n^{th} key (n^{th} key of the m^{th} row). The normalization which has been carried out consists in dividing the values by the mean energy obtained when comparing two different images of the same key. Table 5 shows the final values of normalized minimum energy for the same images when the factors are weighted as in equation (3). As observed when comparing both tables, these values are

$W_{\min}$	$k1:1$	$k1:2$	$k1:3$	$k2:1$	$k2:2$	$k2:3$	$k3:1$	$k3:2$	$k3:3$
$k1:1$	0.0000	1.1385	0.8397	3.7433	4.3798	4.3227	7.2674	6.3000	5.9164
$k1:2$		0.0000	0.5509	5.6977	6.1319	6.0507	8.0686	6.8423	6.6339
$k1:3$			0.0000	4.8914	5.5298	5.3115	7.9518	6.5084	6.3362
$k2:1$				0.0000	0.8658	1.1344	9.3063	9.2261	8.5619
$k2:2$					0.0000	1.1442	9.6234	9.5260	7.0714
$k2:3$						0.0000	9.5537	9.4011	8.9901
$k3:1$							0.0000	1.3674	1.1965
$k3:2$								0.0000	0.7626
$k3:3$									0.0000

Table 5: Normalized minimum weighted energy values for keys in figure 8.

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, which results in a clearer discrimination of the shapes. Normalized energy values around or under 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 8 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.

8 Object fitting and motion analysis

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 saved shape. Firstly, a translation may be required. It can be extracted from the centers of both coupled shapes by subtracting their coordinates. Equations (4) and (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 .

$$x_c = \frac{\sum_{(x,y)\in I} xO(x,y)}{\sum_{(x,y)\in I} O(x,y)} \tag{4}$$

$$y_c = \frac{\sum_{(x,y)\in I} yO(x,y)}{\sum_{(x,y)\in I} O(x,y)} \tag{5}$$

Let $(x_c^1, y_c^1), (x_c^2, y_c^2)$ be the centers of both objects, equation (6) shows the translation

to perform:

$$T = (x_c^2 - x_c^1, y_c^2 - y_c^1) \tag{6}$$

Secondly, we must consider their sizes. As said before, orientation functions have been normalized to make comparisons. However, we must consider actual sizes, which can be extracted from the segmentation of the images in two regions, object and background.

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

The square root of the quotient of both sizes determines the proportional scale we must use, as shown in equation (7):

$$S = \sqrt{\frac{A^1}{A^2}} \tag{7}$$

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.

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

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

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

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

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

$$\theta = \arctan\left(\frac{\sum_{i=1}^{L-1} (x_i^1 y_i^2 - y_i^1 x_i^2)}{\sum_{i=1}^{L-1} (x_i^1 x_i^2 + y_i^1 y_i^2)} \right) \tag{8}$$

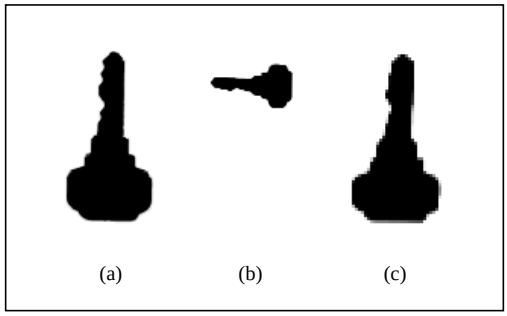

Figure 9: (a) and (b) Input images corresponding to the same key. (c) Adaptation of the second image to the first one by means of translation, rotation and scaling.

With these 3 transformations, given by the parameters in equations (6), (7) and (8), we could adapt the position, orientation and distance of an object to fulfill certain conditions, e.g. to be able to grab it. Figures 9 and 10 show two examples of object fitting with translation, rotation and scaling.

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 respect 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:

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

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 viceversa. Thus, 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 in [OH97].

Figure 11 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

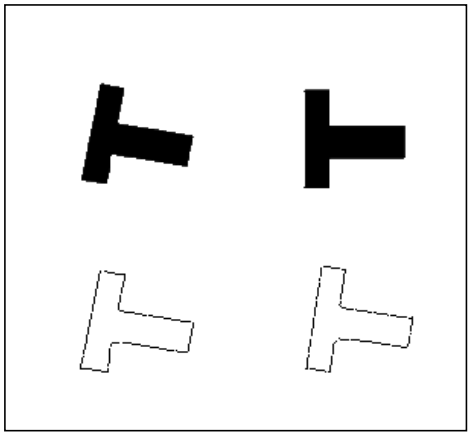

Figure 10: Contour fitting of a couple of consecutive images from a motion sequence with translation, rotation and scaling (The contour of the image on the right is adjusted to image on the left).

Frames	T_x	T_y	θ	S	d
1-2	-12	-6	0.108	1.00	0.36
2-3	-16	-8	0.275	0.97	0.58
3-4	-12	-10	0.212	0.99	0.45
4-5	-7	-19	0.130	1.00	0.41
5-6	-5	-13	0.206	0.99	0.55
6-7	-8	-13	0.209	1.00	0.48

Table 6: Translation (T_x, T_y), rotation angle θ, scaling proportion S and mean distance d for frames in figure 11.

orientation of the object with a vector whose angle corresponds to the rotation of the object. Table 6 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.

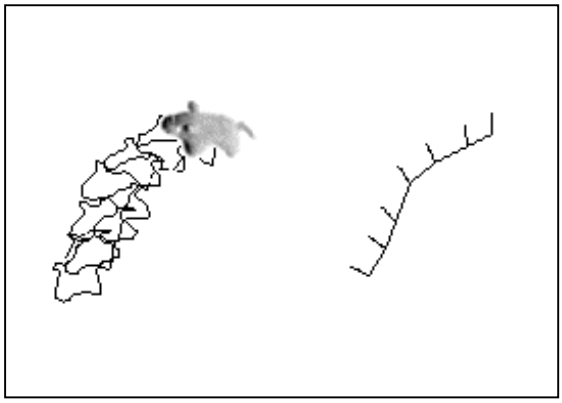

Figure 11: Image sequence and temporal evolution. The trajectory of the object is shown on the right and the vectors represented on it indicate the changes in the orientation from the initial situation.

9 Conclusions

The present article proposes new tools for calculating and identifying edges which are inspired on Newton Filters. Some modifications have been carried out on this set of filters in order to provide them with rotational invariance, but preserving their invariance to global illumination changes.

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 using correlation with a pattern as a characterization method, in such a way that we not only locate the edges, but also identify their 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 tools, it is possible to build orientation functions which can clearly identify shapes and extract global information to fit objects into described patterns. 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 [ZR72], but oriented to polygonal shapes with irregularly spaced series of points, while in our case, signals are discrete, points are equally separated and we use the fast Fourier transform as basic tool for the shape analysis.

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 conservation of information if we consider the location and characterization of changes as the main goal.

Finally, 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.

The previous features show the usefulness of these filters when the objective is not merely the location of edges, but also 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.

References

[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 1333, pp. 406-419. Computer Aided Systems Theory-EUROCAST'97. Springer Verlag, 1997.

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

[HUB88] D.H. Hubel: *Eye, brain, and vision*. New York : Scientific American Library, 1988.

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

[Kan90] E.R. Kandel: *Processing of form and movement in the visual system*. Sensory systems of the brain: sensation and perception, chapter 29, part V, 1990.

[Lon98] S. Loncaric: *A survey of shape analysis techniques*. Pattern Recognition, vol. 31, no. 8, pp. 983-1001, 1998.

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

[OH97] C.F. Olson and D.P. Huttenlocher: *Automatic target recognition by matching oriented edge pixels*. IEEE Transactions on Image Processing, vol. 6, no. 1, pp. 103-113, 1997.

[QAM99] A. Quesada-Arencibia, M. Alemán-Flores and 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, pp. 28-34. Springer Computer Science, 1999.

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

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

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