Computación y Sistemas Vol. 3 No. 1 pp. 5-16
 ISSN 1405-5546 Impreso en México

An Extension to Philips' Algorithm for Moment Calculation

J. H. Sossa-Azuela [1], I. Mazaira-Morales[2] and J. M. Ibarra- Zannatha[2]

[1]Centro de Investigación en Computación-IPN
Unidad Profesional Adolfo López Mateos
Zacatenco, México, D.F. 07738, México
hsossa@cic.ipn.mx
[2]CINVESTAV-IPN
Sección de Control Automático
Av. IPN # 2508, México, D.F. 07300 México

Article received on August 26, 1998; accepted on March 26, 1999

Abstract

A new reformulation of Philips' algorithm for the computation of discrete image moments is presented. As Phillips' method, the new reformulation produces the same exact results but in a faster manner.

Moment contributions due to the presence of holes in the shape (which are not taken into account, to our knowledge, into any boundary based method for moment calculation including Philips') are also introduced into the new method.

Keywords:

Discrete image moments, Green's Theorem, Philips' method, Moment computation, Fast algorithm.

1 Introduction

When an image is discretized at two levels, say 0 and 1, a bilevel or a binary image is obtained. This image is composed of all the flat connected regions representing the projections of the objects onto the discrete plane. From the subsets of points representing these objects several geometrical and topological déscriptors can be obtained to characterize these objects.

Among the geometrical descriptors, the discrete moments have proven to be very important features in many image analysis, pattern recognition and visual inspection applications (Hu, 1962), (Dudani, 1977), (Teague, 1980), (Teh, 1988), (Mingfa, 1989), (Belkasim, 1991), (Prokop, 1992), (Flusser, 1993), (Flusser, 1994), (Flusser, 1996), and (Zhao, 1997). To be really useful, such features need to be computed in a fast manner producing also exact results.

The discrete moments of a binary image $I(x,y)$, defined on the range $0 \le x \le M$, $0 \le y \le N$ are given as

$$m_{pq} = \sum_{x=0}^{M}\sum_{y=0}^{N} x^p y^q I(x,y) \qquad (1)$$

Direct computation of this double sum requires additions and multiplications of order N^2. Thus, the computational cost is too heavy for many applications.

Many algorithms have been developed to speed up the computation of equation 1 by reducing computational redundancy. Fu *et al.* (Fu, 1993) compute the moments from four projections of a binary object. The algorithm uses the 1D Hadamard transform, and is very efficient.

Unfortunately it is limited to the computation of the 10 low order moments.

Another way of reducing the computational redundancy is to compute the moments via the object boundary. The shape of a binary object is totally determined by its boundary. Many boundary based methods have been proposed. Among them the Delta method proposed by Zakaria *et al.* (Zakaria, 1987), for example, assumes that the object under computation is a set of contiguous 1-pixels, lying on lines parallel to the x-axis. The starting position $(x_{s,k}, y_{s,k})$ and the length δ_k of each line is then used to compute the cumulative moments. This makes Zakaria *et al.*'s method only suitable for convex binary shapes. This drawback was recently overcome by Li by using multiple-line-integrals instead of Zakaria et al.'s one-line-segments integrals (Li, 1993). The number of multiplications and additions needed for the computation of m_{pq} by Zakaria *et al.*'s method is reduced from $\varphi(MN)$ to $\varphi(M)$. Unfortunately, an additional $\varphi(MN)$ increment and test operations are needed to determine the numbers δ_k.

Li and Shen (Li, 1991) solve this problem by using a contour tracing algorithm and a discrete approximation of Green's theorem. Depending on the chosen approximation, very different results are obtained. Recently, Philips (Philips, 1993) presented an algorithm to compute the discrete moments of a binary shape using an exact discrete analogous of Green's theorem producing exact results as if the moments were computed by direct summation using equation 1. It also requires fewer calculations than previously exact methods.

Recent results on geometric moment computation for a binary shape in the case of similarity and affine transformations can be found in (Flusser, 1993), (Yang, 1994), (Lin, 1994), (Yang, 1996), (Zhou, 1996), (Zhao, 1997), and (Salama, 1998).

In the next sections a reformulation of Philips' algorithm for the computation of discrete image moments is presented. As Phillips' method the new reformulation produces the same exact results but in a faster manner. Moment contributions due to the presence of holes in the shape (which are not taken into account, to our knowledge, into any boundary based method for moment calculation, including Philips') are also introduced into the new method.

1.1 Paper Outline

The rest of the paper is organized as follows. In section 2 Philips' algorithm is described. The new reformulation along with some guidelines for its practical implementation are presented in section 3. Some experiments showing the advantages of the new method with regard to Phillips' and Hu's methods are presented in section 4. A complexity analysis between the proposed approach, Phillips' and other popular methods is presented in section 5. Finally, in section 6 some conclusions and present work are also given.

2 Philips' Algorithm for the Fast Computation of Discrete Moments

Philips' algorithm development begins with his definition of contour of a discrete region defined in the xy plane.

Definition 1 *Let Ω be a discrete region in the xy plane, the contour, $\partial\Omega$ of a region is defined as those elements in Ω whose right-neighbors (EAST direction) are not in Ω plus those elements not in Ω whose right-neighbors are in Ω.*

Equations 2 to 4 express mathematically this definition:

$$\partial\Omega = \partial\Omega_x^+ \cup \partial\Omega_x^- \tag{2}$$

where

$$\partial\Omega_x^+ = \{(x,y) : (x,y) \in \Omega, (x+1,y) \notin \Omega\} \tag{3}$$

$$\partial\Omega_x^- = \{(x,y) : (x,y) \notin \Omega, (x+1,y) \in \Omega\} \tag{4}$$

Figure 1 shows Philips' contour with regard to Definition 1.

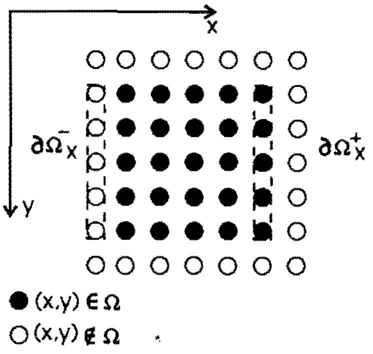

Figure 1: *The contour of an object according to Philips' definition. It is formed by those pixels enclosed by the dotted rectangles.*

2.1 Green's Theorem

Different versions of Green's theorem exist. The one adopted by Philips (Philips, 1993) states that:

Let $f(x,y)$ be any continuous function in the xy plane, then

$$\iint_{\Omega} \frac{\partial}{\partial x} f(x,y)\,dx\,dy = \oint_{\partial\Omega} f(x,y)dy \tag{5}$$

where Ω is a two-dimensional region and $\partial\Omega$ its contour (traversed in the counterclockwise direction).

Following definition 1, the discrete analog of Green's theorem as stated by Philips is given as (Philips, 1993):

$$\sum\sum_{(x,y)\in\Omega} \nabla_x f(x,y) = \sum_{(x,y)\in\partial\Omega_x^+} f(x,y) - \sum_{(x,y)\in\partial\Omega_x^-} f(x,y) \tag{6}$$

where $\nabla_x f(x,y) = f(x,y) - f(x-1,y)$ represents the inferior differential of function $f(x,y)$ with respect to x.

2.2 Philips' Algorithm

Philips uses equations 1 and 6 to derive its method for computing discrete moments by putting $f(x,y) = g(x)y^q$, where $g(x)$ satisfies $\nabla_x g(x) = x^p$. Since $\sum_{n=a}^{b} \nabla_n f(n) = f(b) - f(a-1)$, the equation for $g(x)$ can be rewritten as

$$g(x) = C + \sum_{n=0}^{x} n^p \tag{7}$$

Remember also that if x is a variable, and that if $x^{(k)}$ is a polynomial in x of degree k, then $x^{(k)}$,for $k = 0, 1, 2, ...$ can be expanded to give

$$\begin{aligned} x^{(0)} &= 1 \\ x^{(1)} &= x \\ x^{(2)} &= x^2 - x \\ x^{(3)} &= x^3 - 3x^2 + 2x, \text{ etc.} \end{aligned} \tag{8}$$

which can be written as

$$\begin{pmatrix} x^{(0)} \\ x^{(1)} \\ x^{(2)} \\ x^{(3)} \\ \vdots \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & \\ 0 & 1 & 0 & 0 & 0 & \\ 0 & -1 & 1 & 0 & 0 & \\ 0 & 2 & -3 & 1 & 0 & \cdots \\ \vdots & \vdots & \vdots & \vdots & \vdots & \end{pmatrix} \begin{pmatrix} 1 \\ x \\ x^2 \\ x^3 \\ \vdots \end{pmatrix} \tag{9}$$

The matrix of this equation is called the Stirling matrix of the first kind, for its discover. We symbolize it as S, and it is seen to give $x^{(i)}$ as a polynomial in x, i.e.,

$$x^{(i)} = \sum_{j=0}^{i} [S]_{ij}\, x^j \tag{10}$$

In reference (Morrison, 1969), page 39, it is shown that

$$S_p(x) = \sum_{n=0}^{x} n^p = (x+1) \sum_{j=0}^{p} \frac{[S^{-1}]_{pj}}{j+1} \sum_{v=0}^{j} [S]_{jv}\, x^v \tag{11}$$

where S^{-1} is the inverse of S.

Finally, from equations 1 and 6 and the value of $f(x,y)$ the discrete moments, integrating with respect to x can be directly evaluated as

$$m_{pq} = \sum_{(x,y)\in\partial\Omega_x^+} S_p(x)y^q - \sum_{(x,y)\in\partial\Omega_x^-} S_p(x)y^q \tag{12}$$

For each border pixel and for all required p and q, the value of $S_p(x)y^q$ is computed up to a constant factor then added or subtracted from (depending on whether the pixel lies on $\partial\Omega_x^+$ or on $\partial\Omega_x^-$) an accumulator A_{pq}. Once the whole of $\partial\Omega_x$ has been traced, A_{pq} will contain the values of m_{pq} up to a constant factor.

Table 1 lists the equations needed for the computation of the moments m_{pq} of degree not exceeding 3. They require 9 multiplications and 2 additions per border pixel. For each border pixel, the updating of accumulators requires 10 extra additions while 8 comparisons are needed for the contour tracing. Hence the total complexity of Philips's method is 9 multiplications and 20 additions per border pixel. The method of Zakaria *et al.* requires a maximum of $N + 6$ additions and 25 multiplications for each line of pixels or, since each line corresponds to 2 border pixels, 12.5 multiplications and $N/2 + 3$ additions per border pixel. Philips' algorithm is clearly faster.

Moment	Values
m_{00}	$x+1$
m_{01}	$(x+1)y$
m_{02}	$(x+1)y^2$
m_{03}	$(x+1)y^3$
m_{10}	$\frac{1}{2}(x+1)x$
m_{11}	$\frac{1}{2}(x+1)xy$
m_{12}	$\frac{1}{2}(x+1)xy^2$
m_{20}	$\frac{1}{3}(x+1)x(x+1/2)$
m_{21}	$\frac{1}{3}(x+1)x(x+1/2)y$
m_{30}	$\frac{1}{4}((x+1)x)^2$

Table 1: *Values needed to update the accumulators at each border pixel according to Philips.*

Example 1

In order to illustrate the functioning of Philips' method let us consider the sixteen element binary region shown in Figure 2 and let us outline the computation of its first moments m_{00}, m_{01} and m_{10}. Suppose that the starting point coordinates of the contour are $x = 6$, $y = 4$. Table 2 resumes the followed steps and the partial and final results in the computation of the required moments. One can easily verify that these values are the same as if they were obtained through equation 1.

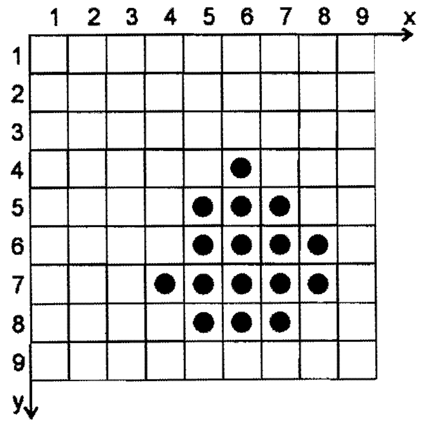

Figure 2: *Binary region of example 1.*

In the next section a reformulation of Philips' algorithm producing the same exact results but in a faster way is proposed. Moment contributions due to the presence of holes in the shape (which are not taken into account into Philips' method) are also introduced into the new method.

3 Proposed Approach

In this section a new reformulation of Philips' method is presented. This new reformulation comprises, in one hand, a diminution in the time of computation of the moments of any planar shape without holes, and, in the other hand, the generalization of its use for a planar object (any shape with and without holes). The proposed modifications are:

1. an extension of the integration process in both axes of the plane, which along with a decision algorithm to determine the membership of a given pixel to the different sections of the contour, it allows to simplify the expressions to compute the moments, and that in lots of cases conveys time saving.

2. incorporate an algorithm allowing the presence of holes in the shape.

The proposed modifications along with some guidelines for the practical implementation of the proposed method are presented next.

3.1 Integration Process Extension

Green's equivalent version in the case of the y axis can be written as follows:

Let $f(x, y)$ any continuous function in the xy plane, then

$$\iint_{\Omega} \frac{\partial}{\partial y} f(x,y)\, dx\, dy = \oint_{\partial\Omega} f(x,y) dx \tag{13}$$

where again Ω is a two-dimensional region and $\partial\Omega$ its contour. This can be done due to symmetry reasons.

In the same way, Philips' contour equivalent definition but with respect to the y axis can be stated as follows:

Definition 2 *Let Ω be a discrete region in the xy plane, the contour, $\partial\Omega$ of a region is defined as those elements in Ω whose inferior-neighbors (SOUTH direction) are not in Ω plus those elements not in Ω whose inferior-neighbors do are in Ω.*

Equations 14 to 16 express mathematically this definition:

$$\partial\Omega = \partial\Omega_y^+ \cup \partial\Omega_y^- \tag{14}$$

where

$$\partial\Omega_y^+ = \{(x,y) : (x,y) \in \Omega, (x, y+1) \notin \Omega\} \tag{15}$$

$$\partial\Omega_y^- = \{(x,y) : (x,y) \notin \Omega, (x, y+1) \in \Omega\} \tag{16}$$

Figure 3 shows the equivalent version of Philips' contour, this time, with regard to definition 2.

The discrete analog of Green's theorem with respect to the y axis is given as:

$$\sum\sum_{(x,y)\in\Omega} \nabla_y f(x,y) = \sum_{(x,y)\in\partial\Omega_y^+} f(x,y) - \sum_{(x,y)\in\partial\Omega_y^-} f(x,y) \tag{17}$$

where now $\nabla_y f(x,y) = f(x,y) - f(x,y-1)$ represents the inferior differential of function $f(x,y)$ with respect to y.

Computation of the discrete moments, integrating with respect to the y axis can be obtained as follows:

$$m_{pq} = \sum_{(x,y)\in\partial\Omega_y^+} S_q(y)x^p - \sum_{(x,y)\in\partial\Omega_y^-} S_q(y)x^p \tag{18}$$

step	x	y	x'	y'	m_{00}	m_{01}	m_{10}
1	6	4	6	4	7	28	42
2	7	5	7	5	7+8=15	28+40=68	42+56=98
3	8	6	8	6	15+9=24	68+54=122	98+72=170
4	8	7	8	7	24+9=33	122+63=185	170+72=242
5	7	8	7	8	33+8=41	185+64=249	242+56=298
6	5	8	4	8	41-5=36	249-40=209	298-20=278
7	4	7	3	7	36-4=32	209-28=181	278-12=266
8	5	6	4	6	32-5=27	181-30=151	266-20=246
9	5	5	4	5	27-5=22	151-25=126	246-20=226
10	6	4	5	4	22-6=16	126-24=102	226-30=196/2=98

Table 2: *Solution to example 1. x and y are the real coordinates of the pixel, x' and y' are the actual coordinates used according to definition 1.*

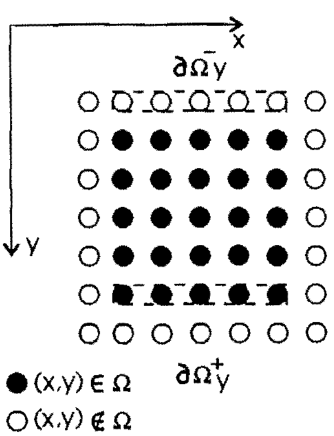

Figure 3: *Equivalent version of Philips' contour.*

Moment	Values
m_{00}	$y+1$
m_{01}	$\frac{1}{2}(y+1)y$
m_{02}	$\frac{1}{3}(y+1)y(y+1/2)$
m_{03}	$\frac{1}{4}((y+1)y)^2$
m_{10}	$(y+1)x$
m_{11}	$\frac{1}{2}(y+1)xy$
m_{12}	$\frac{1}{3}(y+1)y(y+1/2)x$
m_{20}	$(y+1)x^2$
m_{21}	$\frac{1}{2}(y+1)yx^2$
m_{30}	$(y+1)x^3$

Table 3: *Equivalent expressions for the evaluations of the moments according to Definition 2.*

where now

$$S_q(y) = \sum_{n=0}^{y} n^q = (y+1)\sum_{j=0}^{q} \frac{[S^{-1}]_{qj}}{j+1} \sum_{v=0}^{j} [S]_{jv}\, y^v. \tag{19}$$

Again, S is the Stirling matrix of the first kind and S^{-1} its inverse.

Following the same procedure used in the development of Philips' algorithm, but exchanging in this case x by y and vice versa, we get the set of equivalent expressions for the evaluation of the same set of moments in an exact way, but integrating this time with respect to y. Table 3 lists the equations needed for the computation of the moments m_{pq} of degree no exceeding 3. Figure 4 sums up the two definitions of a contour. Sub indices x and y define the axis with regard to the contour which is defined.

From the above discussion, it is clear that the results obtained, either, with respect to x or with respect to y, are the same and exact. Note also from tables 1 and 3 that the expression to compute the same moment is simpler in one direction that in the other. We can thus use the simpler one to compute the corresponding moment without altering the final result obtaining, as a result, an extra time reduction. Column two of table 4 lists the set of expressions chosen for the computation of the moments m_{pq} of degree not exceeding 3. Five moments are obtained with respect to x (according to Definition 1); the other five are obtained with respect to y, i.e. according to Definition 2.

Now, if we analyze Definitions 1 and 2, we note that, in the case of Definition 1, the elements of the negative section of the contour (left side of the objects) match those whose coordinates equal $(x-1,y)$, where (x,y) are the coordinates of the pixel belonging to the real contour of the object. In the same way, for Definition 2, the elements of the negative section of the contour (superior side of the object) match those whose coordinates equal $(x,y-1)$, where (x,y) are the coordinates of the pixel belonging to the real contour of the object. Besides, in both definitions the elements of the positive sections match with those elements of the real contour of the object.

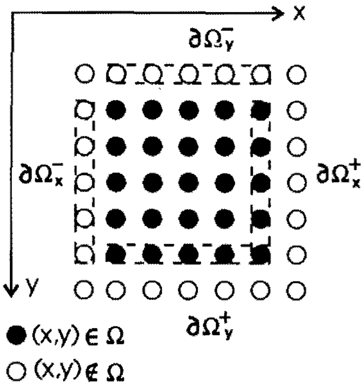

Figure 4: *The objects' contour according to definitions 1 and 2.*

Taking into account this, if the expressions given in column two of table 4 are applied to the negative sections of the contour, the resultant expressions are those shown in column three of table 4, where the values of x and y are now the coordinates of the pixels belonging to the real contour of the object and not to the contour defined according to Definitions 1 and 2. Note that these expressions are much simpler than those used in column 2. This allows to get an extra time saving when the pixel belongs to the negative section of the contour.

Besides, if it happens that a pixel of the contour being analyzed belongs to the positive section and if it has two white horizontal/vertical neighbors, we say that this pixel belongs to both sections of the contour. This fact can be summarized mathematically as follows:

$$\partial\Omega_x^{+/-} = \{(x,y) : (x,y) \in \Omega, (x \pm 1, y) \notin \Omega\} \quad (20)$$

$$\partial\Omega_y^{+/-} = \{(x,y) : (x,y) \in \Omega, (x, y \pm 1) \notin \Omega\} \quad (21)$$

As it was said in section 2 the contribution of each contour pixel to the calculus of the moments is added or subtracted to accumulator A_{pq} depending on if the pixel belongs to the positive or the negative section of the contour, respectively. This makes that for those pixels belonging to both sections of the contour, the total contribution to the moments for a given object be the difference between the expressions given in columns 2 and 3 of table 4. These differences are shown in column 4. An extra time saving is gained.

3.2 Practical Implementation

To practically implement the new algorithm, a method to determine the pixel's membership to the different

Moment	$\partial\Omega_x^+$	$\partial\Omega_x^-$	$\partial\Omega_x^{+/-}$
m_{00}	$x+1$	x	1
m_{01}	$(x+1)y$	xy	y
m_{02}	$(x+1)y^2$	xy^2	y^2
m_{03}	$(x+1)y^3$	xy^3	y^3
m_{12}	$\frac{1}{2}(x+1)xy^2$	$\frac{1}{2}(x-1)xy^2$	xy^2
m_{10}	$(y+1)x$	yx	x
m_{11}	$\frac{1}{2}(y+1)xy$	$\frac{1}{2}(y-1)yx$	yx
m_{20}	$(y+1)x^2$	yx^2	x^2
m_{21}	$\frac{1}{2}(y+1)yx^2$	$\frac{1}{2}(y-1)yx^2$	yx^2
m_{30}	$(y+1)x^3$	yx^3	x^3
	$\partial\Omega_y^+$	$\partial\Omega_y^-$	$\partial\Omega_y^{+/-}$

Table 4: *Column 2: Expressions to be used when the pixel being analysed belongs to the positive section of the contour. Column 3: Expressions to be used when the pixel being analysed belongs to the negative section of the contour. Column 4: Expressions to be used when the pixel being analysed belongs to both sections of the contour. In all cases, the coordinates values to be used are the real ones.*

sections of the object's contour is needed. For this the following definition is necessary.

Definition 3 *Let Ω be a discrete region in the xy plane, and $\partial\Omega$ its contour and let $p \in \partial\Omega$, we call (a_i) the* ***actual direction*** *of p, to the direction from where pixel $p+1$ ($p+1$ belongs also to the same contour) is found, and (a_{i-1}) the* ***anterior direction*** *of p, to the direction from where pixel p was found (see Figure 5 (a) for an example).*

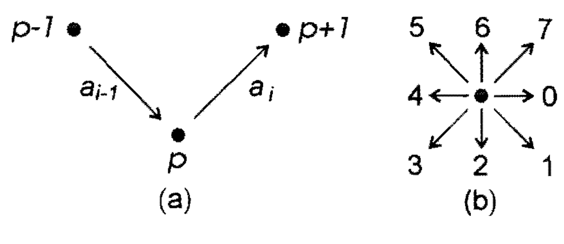

Figure 5: *(a) Anterior and actual directions of pixel p. (b) Directions for 8-directional chain code.*

Anterior and actual directions were also used in (Voss, 1993) to determine if a point is a minimal or a maximal end point of a segment.

As we are working with 8-connected contours, these two directions could be one of the eight directions shown in Figure 5 (b). This allows a direct application to regions represented by their chain code known as Freeman's code (Freeman, 1961). The pixel's membership to the different sections of the contour is determined

1:	if ($p \to 0,7 \to p+1$ and $p-1 \to 0,1,4-7 \to p$)	then $p \Rightarrow \partial\Omega_y^-$
2:	if ($p \to 0,7 \to p+1$ and $p-1 \to 4,5 \to p$)	then $p \Rightarrow \partial\Omega_y^+$
3:	if ($p \to 0,7 \to p+1$ and $p-1 \to 4-7 \to p$)	then $p \Rightarrow \partial\Omega_x^-$
4:	if ($p \to 3,4 \to p+1$ and $p-1 \to 0-5 \to p$)	then $p \Rightarrow \partial\Omega_y^+$
5:	if ($p \to 3,4 \to p+1$ and $p-1 \to 0,1 \to p$)	then $p \Rightarrow \partial\Omega_y^-$
6:	if ($p \to 3,4 \to p+1$ and $p-1 \to 0-3 \to p$)	then $p \Rightarrow \partial\Omega_x^+$
7:	if ($p \to 5,6 \to p+1$ and $p-1 \to 2-7 \to p$)	then $p \Rightarrow \partial\Omega_x^-$
8:	if ($p \to 5,6 \to p+1$ and $p-1 \to 2-5 \to p$)	then $p \Rightarrow \partial\Omega_y^+$
9:	if ($p \to 5,6 \to p+1$ and $p-1 \to 2,3 \to p$)	then $p \Rightarrow \partial\Omega_x^+$
10:	if ($p \to 1,2 \to p+1$ and $p-1 \to 0-3,6,7 \to p$)	then $p \Rightarrow \partial\Omega_x^+$
11:	if ($p \to 1,2 \to p+1$ and $p-1 \to 0,1,6,7 \to p$)	then $p \Rightarrow \partial\Omega_y^-$
12:	if ($p \to 1,2 \to p+1$ and $p-1 \to 6,7 \to p$)	then $p \Rightarrow \partial\Omega_x^-$

Table 5: *Set of rules to determine the pixel's membership.*

through one of the of rules shown in table 5 while the object's contour is followed. From this table, rule 2 would mean, for example, that if pixel $p+1$ is found coming from actual direction a_0 or actual direction a_7 and if pixel p is found coming from anterior directions a_4 or a_5 then assign pixel p to region $\partial\Omega_y^+$. These rules were obtained after a making an analysis of the different possibilities a pixel p can be reached from pixel $p-1$ and pixel p can reach pixel $p+1$. This way the computation of the discrete moments is reduced to the following of the object's contour, determining at each instant for each pixel its belonging section and applying the expressions given in columns 2 to 4 from table 4 according to its membership to update the corresponding moment's values.

It is important to note that if the pixel belongs solely to a contour given by Definition 1 only the first 5 expressions of the corresponding column of table 4 will be evaluated. In the same way, if the pixel belongs solely to a contour given by Definition 2, then only the last 5 expressions of the corresponding column of table 4 will be evaluated. If a pixel belongs to both sections, then all 10 expressions will be applied. Note also that when using table 4 the real coordinates of the pixel must be adopted.

Example 2

In order to illustrate the functioning of the method let us consider the same sixteen element binary region used example 1 (Figure 2) and let us outline the computation of its first moments m_{00}, m_{01} and m_{10}. Suppose again that the starting point's coordinates of the contour are $x=6$, $y=4$. Table 6 resumes the followed steps and the partial and final results in the computation of the required moments. Note that the results obtained by the new method are the same as those provided by Philips in example 1.

3.3 Taking Holes into Account

As mentioned at the beginning of section 3, the new method would incorporate moment contributions due to the presence of holes in the shape. These, as we said are not taken into account for any boundary based method for moment calculation including Philips'.

In this section we derive the expressions needed for the computation of the discrete moments for the general case of a shape with holes. This result can be summarized by the following proposition:

Proposition 1 *Let Ω be a discrete region in the xy plane with n holes, $\partial\Omega$ its external contour and $\partial\Omega_i$ the contour of each cf the its n holes, the final moments of Ω in terms of its external contour and each hole's contour (internal contour) can be obtained either as*

$$\begin{aligned} m_{pq(object)} &= m_{pq(external\ contour)} \\ &\quad - \sum_{i=1}^{n} m_{pq(internal\ contour[i])} \end{aligned} \tag{22}$$

where $m_{pq(\text{external contour})}$ is the contribution to the moment m_{pq} from the external contour and $m_{pq(\text{internal contour}[i])}$ is the contribution to the moment m_{pq} from the internal contour of each of the n holes according to equations 8, 9, 18 and 19.

Proof. Trivial.

The only reference we found where the authors say how to compute the moments of a hole can be found in (Voss, 1993, pp. 233-235). Unlike, this work, we do explain clearly how to use this information to compute the moments of a whole object comprising holes.

To obtain exact results when equation 22 is applied it is necessary to count on a good definition of an internal contour; this with the aim of not including in the computations elements belonging to the object when the contributions due to holes are being computed.

If instead of giving a new definition for an internal contour, we adopt the same definitions already given for

step	x	y	a_i	a_{i-1}	membership	m_{00}	m_{01}	m_{10}
1	6	4	1	7	$\partial\Omega_x^-$, $\partial\Omega_x^+$, $\partial\Omega_y^-$	1	4	-24
2	7	5	1	1	$\partial\Omega_x^+$, $\partial\Omega_y^-$	1+8=9	4+40=44	-24-35=-59
3	8	6	2	1	$\partial\Omega_x^+$, $\partial\Omega_y^-$	9+9=18	44+54=98	-59-48=-107
4	8	7	3	2	$\partial\Omega_x^+$, $\partial\Omega_y^+$	18+9=27	98+63=161	-107+64=-43
5	7	8	4	3	$\partial\Omega_x^+$, $\partial\Omega_y^+$	27+8=35	161+64=225	-43+63=20
6	6	8	4	4	$\partial\Omega_y^+$	35	225	20+54=74
7	5	8	5	4	$\partial\Omega_x^-$, $\partial\Omega_y^+$	35-5=30	225-40=185	74+45=119
8	4	7	7	5	$\partial\Omega_x^-$, $\partial\Omega_y^-$, $\partial\Omega_y^+$	30-4=26	185-28=157	119+4=123
9	5	6	6	7	$\partial\Omega_x^-$	26-5=21	157-30=127	123
10	5	5	7	6	$\partial\Omega_x^-$, $\partial\Omega_y^-$	21-5=16	127-25=102	123-25=98

Table 6: *Solution to example 2.*

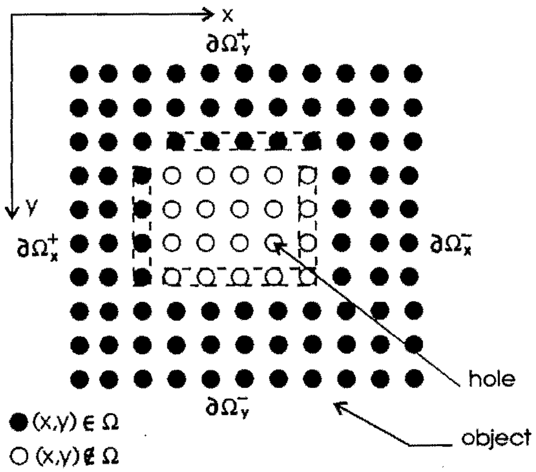

Figure 6: *Internal contour.*

a contour, but this time applied to a hole (see Figure 6), it can be easily shown that equations 9 and 18 become:

$$m_{pq} = \sum_{(x,y)\in\partial\Omega_x^-} S_p(x)y^q - \sum_{(x,y)\in\partial\Omega_x^+} S_p(x)y^q \qquad (23)$$

$$m_{pq} = \sum_{(x,y)\in\partial\Omega_y^-} S_q(y)x^p - \sum_{(x,y)\in\partial\Omega_y^+} S_q(y)x^p \qquad (24)$$

Substituting equations 9, 18, 23 and 24 into equation 22 we get the final expressions for the evaluation of the discrete moments for any planar shape in the plane.

To correctly implement the above method, an algorithm to determine when an internal contour belongs to an object or not is needed. The one proposed in (Mazaira, 1994) is used here. Generally speaking this algorithm scans an input image until a pixel belonging to an external contour is found. The algorithm proceeds to follow the corresponding external contour and computes its contributing moments. Once the initial pixel has been reached, the algorithm continues to scan the image to see if a pixel belonging to an internal contour is found. If this is case, it takes into account the label of the last external contour found, follows the corresponding internal contour and computes the contributing moments updating the initial object's moments. This process is repeated until all objects in the image have been processed. For more details refer to (Mazaira, 1994).

4 Experiments, Performance and Discussion

In this section, we show experimentally that the proposed modifications introduced to Philips' algorithm makes it faster, giving as a result the new algorithm. In this case, the algorithm if of course compared with Philips' method taking as reference Hu's method (Hu, 1962). For this we use the set of objects shown in Figure 7. For each shape the ten moments $m_{pq}, (p+q) \leq 3$, were computed. The results are shown in table 7. We can see that the total time required to compurte the 10 moments by our method, is smaller compared to the other two methods.

Object No	Hu's	Phillip's	New
1	0.257	0.155	0.087
2	0.233	0.141	0.080
3	0.197	0.119	0.067
4	0.222	0.134	0.076
Total time	0.909	0.549	0.310

Table 7: *Time (in seconds) requiered to compute the 10 moments for the four objects shown in figure 7 as computed by Hu's and Phillips' methods and new. In all cases a 133 Mhz Pentium PC computer was used.*

The reduction on time is due to the number of ad-

$$m_{pq(\text{object})} = \sum_{(x,y)\in\partial\Omega_x^+(\text{ext cont})} S_p(x)y^q - \sum_{(x,y)\in\partial\Omega_x^-(\text{ext cont})} S_p(x)y^q - \sum_{i=1}^{n}\left(\sum_{(x,y)\in\partial\Omega_x^-(\text{int cont[i]})} S_p(x)y^q - \sum_{(x,y)\in\partial\Omega_x^+(\text{int cont[i]})} S_p(x)y^q\right) \tag{25}$$

$$m_{pq(\text{object})} = \sum_{(x,y)\in\partial\Omega_y^+(\text{ext cont})} S_q(y)x^p - \sum_{(x,y)\in\partial\Omega_y^-(\text{ext cont})} S_q(y)x^p - \sum_{i=1}^{n}\left(\sum_{(x,y)\in\partial\Omega_y^-(\text{int cont[i]})} S_q(y)x^p - \sum_{(x,y)\in\partial\Omega_y^+(\text{int cont[i]})} S_q(y)x^p\right) \tag{26}$$

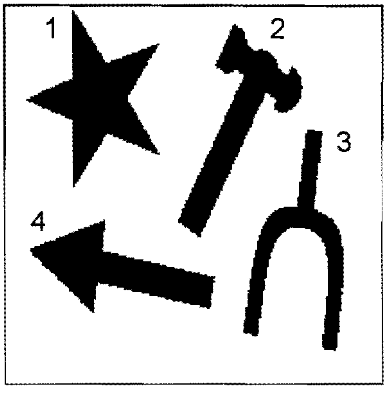

Figure 7: *Set of objects for experiment number 1*

ditions and the number of multiplications implied in the computation of the moments for each pixel using the new method are in general less that in the case of the other methods including Philips' one. This is particularly noticeable if the pixel belongs to the negative section of the contour or to both sections of the contour (section 3.1). This is next explained in more detail in the next section.

5 Complexity Analysis

In this section a complexity analysis between the proposed approach and Phillips' is presented. As it will be shown our method is surpassed only in the case of exactly horizontal enlarged regions, whose probability of occurrence is rather small.

Taking into account just the operations needed to compute the moments, and supposing that n_{x^+}, n_{x^-} and n_{y^+}, n_{y^-} are, respectively, the number of pixels belonging to the positive and negative sections of the shape's contour according to Definitions 1 and 2, we can see that:

- $4 \times (n_{x^+} + n_{x^-}) + 5 \times (n_{y^+} + n_{y^-})$ multiplications, and
- $1 \times (n_{x^+} + n_{x^-}) + 1 \times (n_{y^+} + n_{y^-})$ additions

are needed to the compute the moments.

In the same way, from column 4 of table 4, and supposing that $n_{x^{+/-}}$ and $n_{y^{+/-}}$ are, respectively, the number of pixels belonging to both sections of the contour according to Definitions 1 and 2, one can also see that for those pixels belonging to both sections of the contour, the complexity of the algorithm is reduced to:

- $3n_{x^{+/-}} + 4n_{y^{+/-}}$ multiplications, and
- 0 additions.

Using the same nomenclature, and without taking into acount the 10 extra additions for updating the accumulators and the 8 comparisons needed for contour tracing,.the complexity of the Philips' algorithm is:

- $9(n_{x^+} + n_{x^-})$ multiplications, and
- $2(n_{x^+} + n_{x^-})$ additions.

If both methods are compared, one can conclude that:

- If $(n_{x^+} + n_{x^-}) < (n_{y^+} + n_{y^-})$ then Philips' algorithm is faster than the proposed one.
- If $(n_{x^+} + n_{x^-}) > (n_{y^+} + n_{y^-})$ then Philips' algorithm is slower than the proposed one.
- If $(n_{x^+} + n_{x^-}) = (n_{y^+} + n_{y^-})$ then Philips' algorithm is equal to or slower than the proposed one.

It is obvious from above analysis that the modifications introduced to Phillips' algorithm represent an improvement in relation to the time needed to compute the moments through the object's contour. This improvement is surpassed only in the case of enlarged regions exactly horizontal (see example 3), whose probability of occurrence is, of course, very small.

Example 3: Case where Philips' algoritm is faster than the proposed one.

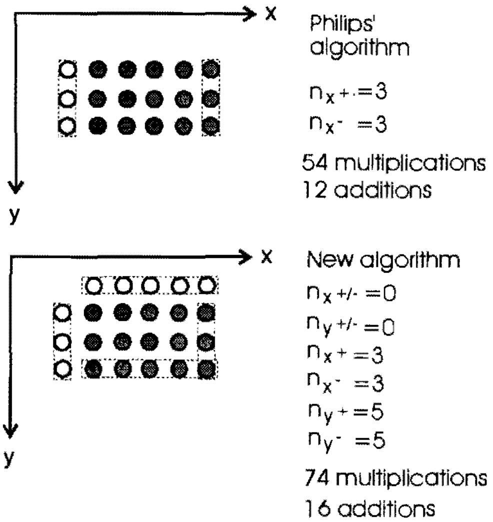

Figure 8: *Case where Philips' algorithm gives better times than the proposed method.*

With respect to other popular methods also using contour information and providing exact results our method is comparable and in some cases faster. Let's suppose that the image has N rows and N columns, and that the object occupies the entire intensity matrix, we have thus an object composed of N by N pixels. Now, if N_v and N_y are the number of pixels of the border of the shape according to Voss and Yang, and M is the number of pixels of the border of the shape and $n = \log_2 N$ according to Fu.

Suppose also for the moment that we do not take into consideration the $10N_y$ additions to cumnpute the Hatamian filtering in the case of Yang's method and the $8M$ and $10N - 13 - n\,(n+6)$ operations to compute the 1D Hadamard Transform in the case of Fu's method. Table 8 lists the number of operations required to compute the first ten moments by our algorihtm and those proposed by Voss (Voss, 1993), Fu (Fu, 1993) and Yang (Yang, 1996). If, for example, that $N = 16$, this is $M = 60, n = 4, N_v = N_y = n_{x+} = n_{x-} = n_{y+} = n_{x-} = 32$, and that a multiplication takes 2 times more time than an addition to be computed, then we have the following results (see table 9). We can see from these two tables that under these considerations our method is slower than Fu's method, as fast as Yang's method but faster than Voss'. However, if we now take into account the number of operations to compute the Hatamian filtering and the 1D Hadamard Transform, we can see in table 10 that our method requires less operations than the other methods. To see this, suppose again that $N = 16$, and that a multiplication takes 2 times more time than an addition to be computed. In table 11 we see the numerical results. Clearly, our method is faster than the other methods. In all cases we did not take into account the number of operations to update the accumulator and to strace the boundary of the shape.

Method	Multiplications	Additions
Voss	$13N_v$	$4N_v$
Fu	17	$2n^3 + 2n^2 + 6n + 23$
Yang	$6N_y$	$8N_y$
New	$4(n_{x+} + n_{x-}) + 5(n_{y+} + n_{y-})$	$(n_{x+} + n_{x-}) + (n_{y+} + n_{y-})$

Table 8: *A first comparison of complexity in computing all the moments of order up to 3 for an object of $N \times N$ pixels.*

Method	Mults.	Additions	Total No. of operations
Voss	416	128	960
Fu	17	207	241
Yang	192	256	640
New	288	64	640

Table 9: *A first comparison in number of operations in computing all the moments of order up to 3 for an object of $N \times N$ pixels.*

Method	Multiplications	Additions
Voss	$13N_v$	$4N_v$
Fu	17	$8M + 10N + 2n^3 + n^2 + 10$
Yang	$6N_y$	$18N_y$
New	$4(n_{x+} + n_{x-}) + 5(n_{y+} + n_{y-})$	$(n_{x+} + n_{x-}) + (n_{y+} + n_{y-})$

Table 10: *A second comparison of complexity in computing all the moments of order up to 3 for an object of $N \times N$ pixels.*

6 Conclusions and Present Research

A new reformulation of Philips' algorithm for the computation of discrete image moments is presented. As Phillips' method the new reformulation produces the same exact results but in faster manner. The only time Philips method gives faster results is in the case of enlarged regions exactly horizontal whose probability of occurrence very small.

Moment contributions due to the presence of holes in the shape (which are not taken into account into any boundary based method for moment calculation including Philips') were also introduced into the new method.

Method	Mults.	Additions	Total No. of operations
Voss	416	128	960
Fu	17	794	828
Yang	192	576	960
New	288	64	640

Table 11: *A second comparison in number of operations in computing all the moments of order up to 3 for an object of $N \times N$ pixels.*

The only situation Philips' method can give faster results than the proposed one is in the case of enlarged regions exactly horizontal. This is a direct consequence of the definition of contour used by Philips where only border pixels to the left and to right of the region are considered in the computations. The probability of occurrence of this situation is, of course, very small.

At the moment, we are working to obtain a more general methodology for the computation of discrete image moments but under general affine transformation which may be used for recognition of affine-deformed objects considering the presence of holes.

Acknowledgments.- The authors would like to thank Dr. H. Ceniceros and the reviewers for their grateful comments to improve the preparation of this paper.

References

S. O. Belkasim and M. Shridhar and M. Ahmadi, "Pattern recognition with moment invariant: A comparative study and new results", *Pattern Recognition*, 24(12):1117-1138, 1991.

S. Dudani and K. J. Breeding and R. D. McGhee, "Aircraft identification by moment invariants", *IEEE Trans. on Computers*, 26:39-45, 1977.

J. Flusser amd T. Suk, "Pattern recognition by affine moment invariants", *Pattern Recognition*, 26(1):167-174, 1993.

J. Flusser and T. Suk, "Affine invariants: a new tool for character recognition", *Pattern Recognition Letters*, 15:433-436, 1994.

J. Flusser and T. Suk, "Recognition of blurred images by the method of moments", *IEEE Trans. on Image Processing*, 5(3):533-538, 1996.

H. Freeman, "On the encoding of arbitrary geometric configurations", *IEEE Trans. on Electronic Computers*, 10:260-268, 1961.

C. W. Fu and J. C. Yen ans S. Chang, "Calculation of moment invariants via Hadamard transform", *Pattern Recognition*, 26(2):287-294, 1993.

M. K. Hu, "Visual pattern recognition by moment invariants", *IRE Trans. on Information Theory*, 8:179-187, 1962.

B. C. Li and J. Shen, "Fast algorithm of moment computation", *Pattern Recognition*, 24:807-813, 1991.

B. C. Li, "A new computation of geometric moments", *Pattern Recognition*, 26:109-113, 1993.

W. G. Lin and Sh. Sh. Wang, "A note on the calculation of moments", *Pattern Recognition Letters*, 15:1065-1070, 1994.

I. Mazaira-Morales, *Manipulating moving objects with an UNIMATE S-103 robot using visual data*, CINVESTAV - IPN, Departamento de Ing. Eléctrica, Sección de Control Automático, Nov, 1994, (In Spanish).

Z. Mingfa and S. Hasani and S. Bhattarai and H. Sing, "Pattern recognition with moment invariants on a machine vision system", *Pattern Recognition Letters*, 9:175-180, 1989.

N. Morrison, *Introduction to Sequential Smoothing and Prediction*, McGraw-Hill, New York, 1969.

W. Philips, "A new fast algorithm for moment computation", *Pattern Recognition*, 26(11):1619-1621, 1993.

R. J. Prokop and A. P. Reeves, "A survey of moment-based techniques for unoccluded object representation and recognition", *CVGIP: Graphical Models and Image Processing*, 54(5):438-460, 1992.

G. I. Salama and L. Abbott, "Moments invariants and quatization effects", *Proc. of the 1998 Int. Conf. on Computer Vision and Pattern Recognition*, 1998, pp. 157-163.

M. R. Teague, "Image analysis via the general theory of moments", *J. Opt. Soc. Am.*, 70(8):920-930, 1980.

C. H. Teh and R. T. Chin, "On image analysis by the method of moments", *IEEE Trans. Pattern Analysis Mach. Intell.*, 10(4):496-513, 1988.

K. Voss, *Discrete Images, Objects, and Functions on Z^n*, Springer-Verlag, Berlin, 1993.

L. Yang and F. Albregtsen, "Fast computation of invarinat geometric moments: a new method giving correct results", *Proc. of the 12th Int. Conf. on Pattern Recognition*, 1999, pp. 201-204.

L. Yang and F. Albregtsen, "Fast and exact computation of Cartesian geometric moments using discrete Green's theorem", *Pattern Recognition*, 29(7):1061-1073, 1996.

M. F. Zakaria and L. J. Vroomen and P. J. Zsombor-Murray and J.M. Van Kessel, "Fast algorithm for the computation of moment invariants", *Pattern Recognition*, 20(6):639-643, 1987.

D. Zhao and J. Chen, "Affine curve moment invariants for shape recognition", *Pattern Recognition*, 30(6):895-901, 1997.

F. Zhou and P. Kornerup, "Computing moments by preffix sums", *Proc. of the 1996 Int. Conf. on Image Processing*, 1996, pp. 619-622.

Juan Humberto Sossa Azuela *received the B.S. Degree in Electronics Engineering from the Universidad de Guadalajara, México in 1981, the M.Sc. degree in Electrical Engineering from the Centro de Investigación y Estudios Avanzados del IPN de México (CINVESTAV-IPN), in 1987 and the Ph.D. degree in Informatics from the Institut National Polytechnique de Grenoble, France in 1992. He is a professor at CIC-IPN. His research interests include computer vision, image processing and control of electromechanical systems using visual data.*

Israel Mazaira Morales *received the B.S. Degree in Automatic Control Engineering from the Universidad de Oriente, Cuba in 1991, and the M.Sc. degree in Electrical Engineering from the Centro de Investigación y Estudios Avanzados del IPN de México (CINVESTAV-IPN), in 1994. He is currently a Ph.D. student there. His research interests include non-linear system theory and control of mechanical manipulators.*

Juan Manuel Ibarra Zannatha *received the B.S. Degree in Electronics Engineering from the Instituto Politécnico Nacional, México in 1973, the M.Sc. degree in Electrical Engineering from the Centro de Investigación y Estudios Avanzados del IPN de México (CINVESTAV-IPN), in 1976 and the Ph.D. degree in Engineering from the Institut de Recherche en Informatique et Systèmes Aleatoires de Rennes, France in 1982. Actually, he is a professor at CINVESTAV-IPN. His research interests include control of mechanical manipulators, neural networks and computer vision.*