



3-D Cartesian Geometric Moment Computation using Morphological Operations and its Application to Object Classification

H. Sossa-Azuela¹, F.J. Cuevas², C. Aguilar-Ibañez¹ and H. Benítez-Muñoz¹

¹Centro de Investigación en Computación del IPN, México DF

²Centro de Investigaciones en Óptica, A. C., León, Guanajuato

E-mails: hsossa@cic.ipn.mx, fjcuevas@cio.mx, caguilar@cic.ipn.mx, hbenitez75@hotmail.com

(Recibido: agosto de 2005; aceptado: septiembre de 2006)

Resumen

Los momentos geométricos tridimensionales son rasgos importantes para el reconocimiento de objetos 3-D y la descripción de forma. El cálculo de estos rasgos en el caso 3-D mediante el método tradicional requiere de una gran número de operaciones. Varios autores han propuesto métodos para su cálculo. La mayoría requieren cómputos de orden N^3 , suponiendo que el objeto es representado como una imagen voxelizada de $N \times N \times N$ elementos. Recientemente, Yang et al. (1996), presenta un método que requiere el cálculo de $O(N^2)$ al usar el teorema discreto de la divergencia que permite calcular la suma de una función para una región discreta n-dimensional mediante la suma sobre una región discreta encerrando al objeto. En este artículo presentamos un nuevo método para el cálculo de momentos 3-D. Para esto, primeramente descomponemos una región en un conjunto de cubos. Esta descomposición forma una partición. Las sumatorias triples usadas en el cálculo de los momentos son reemplazadas por la suma de los momentos de cada cubo de la partición. Los momentos de cada cubo pueden ser calculados en términos de un conjunto muy sencillo de expresiones usando el centro del cubo y su radio.

Mostramos que una vez que la partición ha sido obtenida, el cálculo de los momentos al usar la propuesta es mucho más rápida que la proporcionada por métodos anteriores; la complejidad de la propuesta es de $O(N)$. También mostramos varios ejemplos donde los momentos derivados pueden ser usados en el cálculo de invariantes para el reconocimiento de objetos tridimensionales.

Descriptor: Momentos tridimensionales, cálculo de momentos geométricos, rasgos invariantes, reconocimiento de objetos.

Abstract

Three-dimensional Cartesian geometric moments are important features for 3-D object recognition and shape description. Computing these features in the 3-D case by a straightforward method requires a large number of operations. Several authors have proposed fast methods to compute the 3-D moments. Most of them require computations of order N^3 , assuming that the object is represented by a $N \times N \times N$ voxel image. Recently, Yang et al. (1996) presented a method requiring computation of $O(N^2)$ by using a discrete divergence theorem that allows to compute the sum of a function over an n-dimensional discrete region by a summation over the discrete surface enclosing the object. In this paper, we present a new method to compute 3-D moments. For this, we first decompose the region into a set of balls (cubes) under division. This decomposition forms a partition. Triple summations used in the

computation of the moments are replaced by the sum of the moments of each cube of the partition. The moments of each cube can be computed in terms of a set of very simple expressions using the center of the cube and its radius. We show that once the partition is obtained, moment computation using the proposed approach is much faster than earlier methods; its complexity is in fact of $O(N)$. We also show several experiments where the derived moments can be used to compute invariants useful in the recognition of three-dimensional objects.

Keywords: 2-D geometric moments, 3-D geometric moments, mathematical morphology, metric spaces.

Introducción

The two-dimensional Cartesian geometric moment (for short 2-D moment) of a 2-D object R is defined as Hu (1962):

$$m_{pq} = \iint_R x^p y^q f(x, y) dx dy \quad (1)$$

where $f(x, y)$ is the characteristic function describing the intensity of , and $p+q$ is the order of the moment. Similarly, the three-dimensional Cartesian geometric moment (for short 3-D moment) of order $p+q+r$ of a 3-D object is defined as Lo and Don (1989):

$$m_{pqr} = \iiint_R x^p y^q z^r f(x, y, z) dx dy dz \quad (2)$$

where R is a 3-D region. In the case of a discrete binary 3-D image, the moment of a 3-D homogeneous object represented by voxels is often evaluated as:

$$m_{pqr} = \sum \sum \sum_R x^p y^q z^r \quad (3)$$

with $(x, y, z) \in \mathbf{Z}^3$ and $p, q, r = 0, 1, 2, \dots$

2-D moments are important shape features of a 2-D object, and have been widely

used in image analysis. Applications of 2-D moments can be found in edge detection (Reeves *et al.*, 1983), texture analysis (Albregtsen *et al.*, 1995), movement estimation (Pei and Liou, 1994), image alignment (Flusser and Suk, 1994), object description (Yang *et al.*, 1995) and object recognition (Dudany *et al.*, 1977) and (Flusser and Suk, 1993). Due to their usefulness lots of efforts have been proposed to reduce the time of computation. Among the most important works we can mention the works of Zakaria *et al.* (1997), Li y Shen (1991), Jiang and Bunke (1991), Li (1993), Fu *et al.* (1993), Philips (1993), Yang *et al.* (1994 and 1996) and Sossa *et al.* (1999).

The world around us is three-dimensional by nature. 3-D shape information for an object can be obtained by means of computer tomographic reconstruction, passive 3-D sensors, and active range finders. Like the 2-D moments, 3-D moments have been used in 3-D image analysis tasks including movement estimation (Pei and Liou, 1994), shape estimation (Shen and Li, 1993) , and object recognition (Lo and Don, 1989).

The use of 3-D moments is limited due to computational complexity. To compute all moment of order $p+q+r \leq K$, a straightforward method needs additions and multiplications of $O(K^3 N^3)$ (assuming that the object is represented by an $N \times N \times N$ voxel image).

Some fast methods have been proposed to reduce the computational complexity. In Li (1993), Li uses a polyhedral representation of the object for the computing of its 3-D moments. The number of required operations is a function of the number of edges of the surfaces of the polyhedral. The methods of Cyganski *et al.* (1988), Li and Shen (1992) and Li and Ma (1994) use a voxel representation of the object. The difference among these methods is the way to compute the moments. Cyganski *et al.* (1988) make use of the filter proposed in Budrikis and Hatamian, 1984). Li and Shen use a transformation based on Pascal triangle for the computation of the monomials; only additions are used for the computation of the moments. On the other hand, Li and Ma (1994) relate 3-D moments with the so-called LT moments that are easier to evaluate. Although these methods allow to reducing the number of operations to compute the moments, they require a computation of $O(N^3)$. Recently, Yang *et al.* (1997) presented a discrete divergence theorem to compute the 3-D moments of an object. It allows a reduction in the number of operations to $O(N^2)$. This theorem allows to computing the sum of a function over an n -dimensional discrete region by a summation over the discrete surface enclosing the region.

In this paper we present a method to compute the 3-D moments of a binary region in \mathbf{Z}^3 . The object is first partitioned into convex balls which moment evaluation can be reduced to the computation of very simple formulae instead of using triple integrals. The desired 3-D moments are obtained as the sum of the moments of each ball of the partition, given that the intersection among balls is empty. A first effort in the 2-D case was first presented in Sossa *et al.* (2001).

The paper is organized as follows. Basic knowledge for better understanding of the paper is given in section 2. The steps of the proposed methodology are deeply explained in section 3. Some experimental results and some final comments are given in Sections 4 and 5.

Basic Background

This section presents the basic concepts needed to follow the lecture of the paper. Here the words image and function are used as synonyms; the word volume will be used as synonym of the subset of the image domain. The symbol \mathbf{X} will denote an n -dimensional discrete space being a subset of the n -dimensional real space. Most of the times we will work onto a finite subset of (the volume of the integers).

Metric and erosions

Definition 1. A function $d: \mathbf{X} \rightarrow \mathbf{R}^+$ is called a metric (or distance) iff for all $x, y, z \in \mathbf{X}$, it holds that:

- a) $d(x, y) = 0 \Leftrightarrow x = y$
- b) $d(x, y) = d(y, x)$
- c) $d(x, y) + d(y, z) \geq d(x, z)$

Definition 2. The distance function between two points $p, q \in \mathbf{X}$, is defined as

$$d_{\infty}(p, q) = \sum_{i=1}^n |p_i - q_i| \quad (4)$$

is called ∞ -metric.

Definition 3. The pair (\mathbf{X}, d) , where d is a metric is called a metric space.

Definition 4. Given a metric space (\mathbf{X}, d) , the set defined by:

$$B_p^t = \{x \in \mathbf{X} \mid d(x, p) \leq t\} \quad (5)$$

is called a closed ball of radius t with center in $p \in \mathbf{X}$.

Definition 5. Let B be a subset of \mathbf{X} and $p \in \mathbf{X}$, the translation of B by p is defined as:

$$B_p = \{a + p \mid a \in B\} \tag{6}$$

Definition 6. Let A and B subsets of \mathbf{X} , the erosion of A by B denoted by $A \ominus B$, is defined as:

$$A \ominus B = \{x \mid B_x \subseteq A\} \tag{7}$$

The methodology

Moments of a ball in d_∞ metric

The main idea behind the proposed methodology consist of:

1. Decompose the given shape into a union of disjoint balls; we do this by iteratively eroding the shape of interest by means of the method next described in section method based on iterated erosions to get the partition.
2. Compute the geometric moments for each of these balls, and
3. Obtain the final moments as a sum of the moments computed for each ball

As mentioned before, a first effort in the 2-D case was first presented in Sossa *et al.* (2001). Clearly, the process involved to compute the moments of the balls will be simpler and cheaper (in time and resources), as the ball structure is simpler. The d_∞ metric has been chosen as it allows us to generate some of the most-simple balls (cubes in a discrete Cartesian plane).

Before continuing we need to derive the set of expressions that will allow us to compute the geometric moments for each cube using the d_∞ metric in terms of their

radius and center. To obtain this set of expressions, let us consider a cube centered (X_c, Y_c, Z_c) with radius t and coordinates of its vertices in

- $(X_c - t, Y_c - t, Z_c - t),$
- $(X_c + t, Y_c - t, Z_c - t),$
- $(X_c - t, Y_c + t, Z_c - t),$
- $(X_c - t, Y_c - t, Z_c + t),$
- $(X_c + t, Y_c + t, Z_c - t),$
- $(X_c + t, Y_c - t, Z_c + t),$
- $(X_c - t, Y_c + t, Z_c + t),$ and
- $(X_c + t, Y_c + t, Z_c + t),$

Let us also consider well-known Bernoulli's formulation (Yang and Albrechtsen, 1994-1996):

$$\sum_{k=1}^{k=n} k^p = \frac{n^{p+1}}{p+1} + \frac{1}{2} n^p + \frac{B_1 p n^{p-1}}{2!} - \frac{B_2 (p-1)(p-2)n^{p-3}}{4!} + \dots \tag{8}$$

where the last term of the series contains n or n^2 , depending on if p is even or odd respectively. The B_j 's denote Bernoulli's numbers (Yang and Albrechtsen, 1994-1996).

The sum of the powers in direction x for a cube R , can be found by means of equation (8). The limits of the summation are: $X_c - t$ and $X_c + t$, thus:

$$\begin{aligned}
 \sum_{k=X_c-t}^{X_c+t} k^p &= \sum_{k=1}^{X_c+t} k^p - \sum_{k=1}^{X_c-t-1} k^p \\
 &= \left[\frac{(X_c+t)^{p+1}}{p+1} + \frac{1}{2}(X_c+t)^p + \frac{B_1 p (X_c+t)^{p-1}}{2!} - \frac{B_2 (p-1)(p-2)(X_c+t)^{p-3}}{4!} + \dots \right] \\
 &\quad - \left[\frac{(X_c-t-1)^{p+1}}{p+1} + \frac{1}{2}(X_c-t-1)^p + \frac{B_1 p (X_c-t-1)^{p-1}}{2!} - \dots \right]
 \end{aligned} \tag{9}$$

Analogously, for all coordinates in directions *y* and *z* we have that:

$$\begin{aligned}
 \sum_{k=Y_c-t}^{Y_c+t} k^p &= \sum_{k=1}^{Y_c+t} k^p - \sum_{k=1}^{Y_c-t-1} k^p \\
 &= \left[\frac{(Y_c+t)^{p+1}}{p+1} + \frac{1}{2}(Y_c+t)^p + \frac{B_1 p (Y_c+t)^{p-1}}{2!} - \frac{B_2 (p-1)(p-2)(Y_c+t)^{p-3}}{4!} + \dots \right] \\
 &\quad - \left[\frac{(Y_c-t-1)^{p+1}}{p+1} + \frac{1}{2}(Y_c-t-1)^p + \frac{B_1 p (Y_c-t-1)^{p-1}}{2!} - \dots \right]
 \end{aligned} \tag{10}$$

and

$$\begin{aligned}
 \sum_{k=Z_c-t}^{k=Z_c+t} k^p &= \sum_{k=1}^{k=Z_c+t} k^p - \sum_{k=1}^{k=Z_c-t-1} k^p \\
 &= \left[\frac{(Z_c+t)^{p+1}}{p+1} + \frac{1}{2}(Z_c+t)^p + \frac{B_1 p (Z_c+t)^{p-1}}{2!} - \frac{B_2 (p-1)(p-2)(Z_c+t)^{p-3}}{4!} + \dots \right] \\
 &\quad - \left[\frac{(Z_c-t-1)^{p+1}}{p+1} + \frac{1}{2}(Z_c-t-1)^p + \frac{B_1 p (Z_c-t-1)^{p-1}}{2!} - \dots \right]
 \end{aligned} \tag{11}$$

When applying equations (9) to (11) to equation (8), we obtain the expressions for the moments of order $p+q+r$ for a cube *R* of radius *t* centered at (X_c, Y_c, Z_c) . For example, expression for moment m_{000} is:

$$m_{000} = \sum_{i=X_c-t}^{i=X_c+t} 1 \sum_{j=Y_c-t}^{j=Y_c+t} 1 \sum_{k=Z_c-t}^{k=Z_c+t} 1 = (2t+1)(2t+1)(2t+1) = (2t+1)^3. \tag{12}$$

The reaser can easily show that:

$$m_{102} = \frac{m_{000}}{3} X_c(3Z_c^2 + t(t+1)) \quad (12o)$$

$$m_{100} = m_{000} X_c \quad (12a)$$

$$m_{201} = \frac{m_{000}}{3} Z_c(3X_c^2 + t(t+1)) \quad (12p)$$

$$m_{010} = m_{000} Y_c \quad (12b)$$

$$m_{012} = \frac{m_{000}}{3} Y_c(3Z_c^2 + t(t+1)) \quad (12q)$$

$$m_{001} = m_{000} Z_c \quad (12c)$$

$$m_{200} = \frac{m_{000}}{3} (3X_c^2 + t(t+1)) \quad (12d)$$

$$m_{021} = \frac{m_{000}}{3} Z_c(3Y_c^2 + t(t+1)) \quad (12r)$$

$$m_{020} = \frac{m_{000}}{3} (3Y_c^2 + t(t+1)) \quad (12e)$$

$$m_{111} = m_{000} X_c Y_c Z_c \quad (12s)$$

$$m_{002} = \frac{m_{000}}{3} (3Z_c^2 + t(t+1)) \quad (12f)$$

Method based on iterated erosions to get the partition

The following method to compute the geometric moments of a 3-D object $R \subset Z^3$, using morphological erosions is a direct extension to the one described in Sossa *et al.* (2001). It is composed of the following steps:

$$m_{110} = m_{100} Y_c = m_{000} X_c Y_c \quad (12g)$$

1. Initialize 20 accumulators $C_i=0$, for $i=1,2,\dots,20$, one for each geometric moment.

$$m_{101} = m_{100} Z_c = m_{000} X_c Z_c \quad (12h)$$

2. Make $A=R$ and, $B=\{(\pm a, \pm b, \pm c) | a, b, c \in \{-1, 0, 1\}\}$, B is a $3 \times 3 \times 3$ pixel neighborhood in Z^3 .

$$m_{011} = m_{001} Y_c = m_{000} Y_c Z_c \quad (12i)$$

$$m_{300} = m_{000} X_c (X_c^2 + t(t+1)) \quad (12j)$$

3. Assign $A \leftarrow A \ominus B$ iteratively until the next erosion results in \emptyset (the null set). The number of iterations of the erosion operation before set \emptyset appears, is the radius t of the maximal cube completely contained in the original region R . The center of this cube is found in set A just before set \emptyset appears.

$$m_{030} = m_{000} Y_c (Y_c^2 + t(t+1)) \quad (12k)$$

$$m_{003} = m_{000} Z_c (Z_c^2 + t(t+1)) \quad (12l)$$

$$m_{120} = \frac{m_{000}}{3} X_c (3Y_c^2 + t(t+1)) \quad (12m)$$

4. Select one of the points of A and given that the radius t of the maximal cube is known, we use the formulae

$$m_{210} = \frac{m_{000}}{3} Y_c (3X_c^2 + t(t+1)) \quad (12n)$$

derived in the last section to compute the moments of this maximal cube, the resulting values are added to the respective 20 accumulator, C_i for $1,2,\dots,20$.

5. Eliminate this ball from region R , and assign this new set to R .
6. Repeat steps 2 to 5 with the new until it becomes \emptyset .

The method just described gives us as a result the true values of the geometric moments of order $(p+q+r) \leq 3$, using only erosions and the formulae developed in Section of the moments of a ball in d_∞ metric.

By their nature, the erosions can be done in a massively parallel computer in pretty short processing times. This method is, however, a brute force method (BFM). A considerable enhancement can be obtained if steps 4 and 5 are replaced by:

1. Select those points in A at a distance among them greater than $2t$ and use the formulae given by Proposition 1, to compute the geometric moments of these maximal cubes, and add these values to the respective accumulators.
2. Eliminate the maximal cubes from region, and assign this new set to.

The enhanced method (EM) consists in processing all maximal cubes of the same radius in just a step, coming back to the iterated erosions until the value of the radio t should be changed. At this point it is very important to verify that the eliminated cubes do not intersect with those just eliminated, because one of the important conditions is that the set of maximal cubes forms a partition of the image. Thus one has to guarantee that these maximal cubes be disjoint sets.

Experimental Results

Moment computation

The method introduced in this paper is not designed to work on a conventional computer. Experiments were however done on a 233 MHz PC based system. This way, the processing times are only significant when comparing the method eliminating a cube at the time against the method eliminating at the same time all the non-intersecting maximal cubes at the same step.

Both methods were tested on several hundreds of images. All of them are binary and $101 \times 101 \times 101$ pixel sized. These images were obtained by generating at random P touching and overlapping cubes of different sizes inside the $101 \times 101 \times 101$ image. At the beginning all the locations of the $101 \times 101 \times 101$ cube are zero.

The BFM takes on average, over the whole set of images, 320 seconds to compute all moments of order $(p+q+r) \leq 3$. The 320 seconds include the time to compute the partition iteration by iteration. The EM requires only about 80 seconds onto 233 Mhz PC based system to compute the same moments. Again the 80 seconds include the time to get the partition. In both cases most of the time is required to obtain the necessary partitions.

Efficiency of the computation

With respect to other methods providing the same results as if equation 3 were used, our method is faster, once the partition is obtained. As you can appreciate its complexity is of \cdot . To show this, let us suppose that the image has N rows in all the three directions, and that the object occupies the entire intensity volume, we have thus an object composed of voxels, with t its radio. Table 1 lists the number of operations required to compute the first 20 moments by

the straightforward method, by those proposed by Cyganski *et al.* (1988), Li and Shen (1992), Li and Ma (1994) and ours. The computational complexity of the earlier methods shown in table 1 was taken from Yang *et al.* (1997).

For a given t , our method requires only 36 multiplications and 8 additions to compute the 20 moments. To get these two numbers we just added the number of multiplications and additions required by each of the 20 moments to be computed. requires, for example, 3 multiplications and 1 addition. $m_{100} = m_{000} X_c$ requires 1 multiplication and no additions because it is supposed that the term $(2t + 1)^3$ was already computed.

The careful reader can rapidly see from this table that our method is faster than others, even for a small N . The interested reader can easily verify that for greater values of N our method still requires less time. This is due mainly to the fact that our method uses t instead of N to compute the desired moments.

Object Recognition

3-D moments as 2-D moments have been used in object recognition. In many cases those moments are not used in their standard form, this is, directly. They are combined some way to obtain quantities a lit bit changing before set of transformations. We are talking about the so-called *invariant moments*

In the bi-dimensional case the well-known Hu moments, invariant to translations, rotations and scale changes were derived and used since 30 years (Hu, 1962). In Flusser and Suk (1993) and in Reiss (1991) arrived to the other set of invariants, but before general transformations (affine transformations). In this section we present a set of experiments to test performance of invariants derived from the set of the formulae proposed in section moments of a ball in metric d_{∞} .

Several invariants can be derived by means of the Fundamental Theorem of Moment Invariants (FTMI), proposed in Sadjadi and Hall (1980). Two of them are:

$$I_1 = \frac{J_1^2}{J_2} \tag{13}$$

$$I_2 = \frac{\Delta_2}{J_1^3} \tag{14}$$

with

$$j_1 = \mu_{200} + \mu_{020} + \mu_{002} \tag{15}$$

$$J_2 = \mu_{020}\mu_{002} - \mu_{001}^2 + \mu_{200}\mu_{002} - \mu_{101}^2 + \mu_{200}\mu_{020} - \mu_{110}^2$$

$$\Delta_2 = \det \begin{bmatrix} \mu_{200} & \mu_{110} & \mu_{101} \\ \mu_{110} & \mu_{020} & \mu_{011} \\ \mu_{101} & \mu_{011} & \mu_{002} \end{bmatrix}$$

The μ_{pqr} are computed as:

$$\mu_{pqr} = \sum \sum \sum (x - \bar{x})^p (y - \bar{y})^q (z - \bar{z})^r \tag{15}$$

with

$$\bar{x} = \frac{m_{100}}{m_{000}}, \bar{y} = \frac{m_{010}}{m_{000}}, \bar{z} = \frac{m_{001}}{m_{000}}$$

In this section we show how the invariants moments obtained through the methodology proposed in this paper can be used to differentiate among 3-D objects. In the experiments were used the three objects shown in figure 1: A sphere of radius of 8 voxels, a pyramid with base of 25x25 voxels

and a parallelepiped of $55 \times 13 \times 13$ voxels. Figure 2 shows three transformed versions (translated, rotated and scaled) of the objects of figure 1. Table 2 shows the values of the corresponding transformations for each object. Table 3 shows the values of the invariant moments for each object's image. In the Table the average times in seconds invested for the computation of the invariants; this includes the time used to obtain the corresponding partitions.

Figure 3, shows the values of the computed invariants for each object (sphere: , pyramid: and parallelepiped:). From this figure and table 3 the reader can appreciate, in the one hand, that except for class (sphere), the values obtained for the other two classes (pyramid and parallelepiped) change very little inside each class. In the other hand, the differences in the invariants from class to class suggest that any linear classifier could help to differentiate among the three objects used.

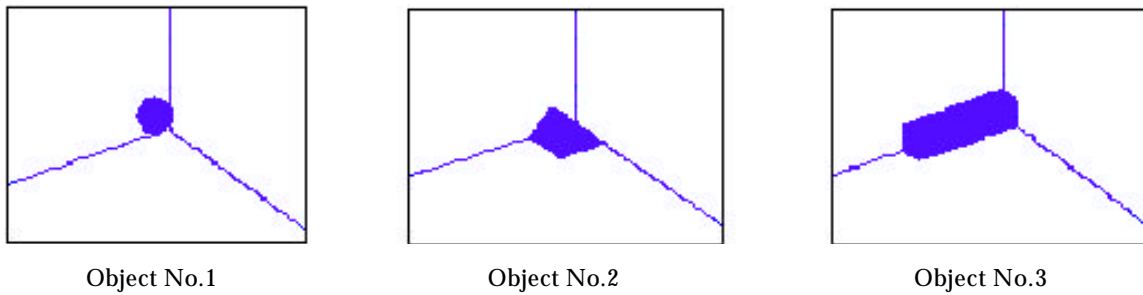


Figure 1. The three objects used in the experiments

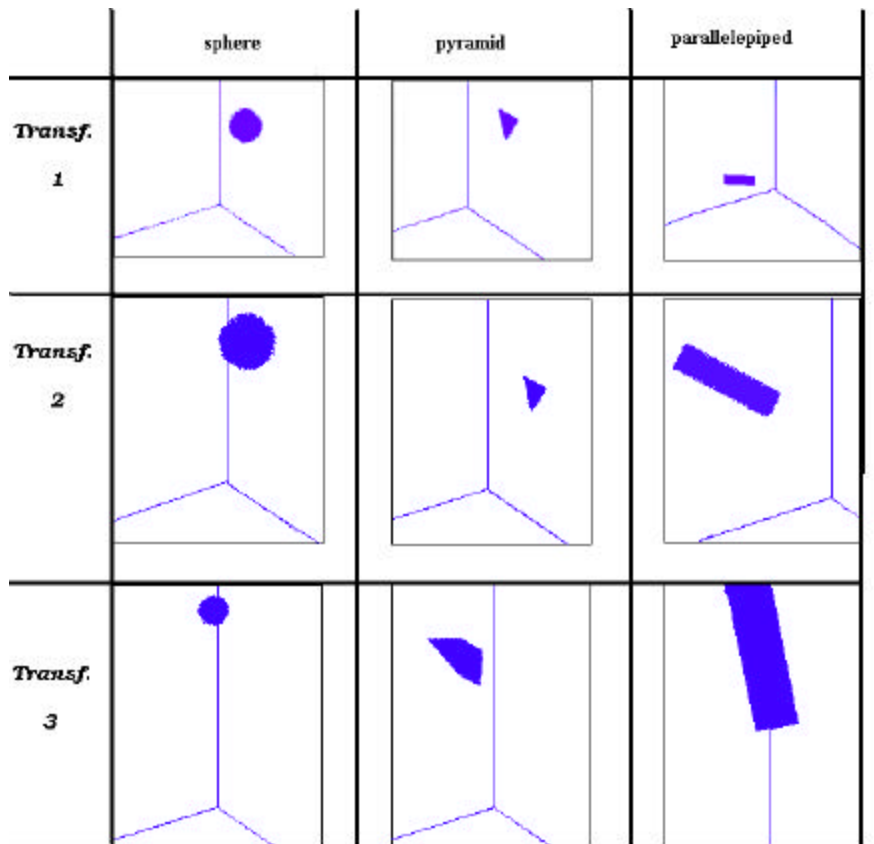


Figure 2. Three transformed versions (translated, rotated and scaled) of the objects of Figure 1

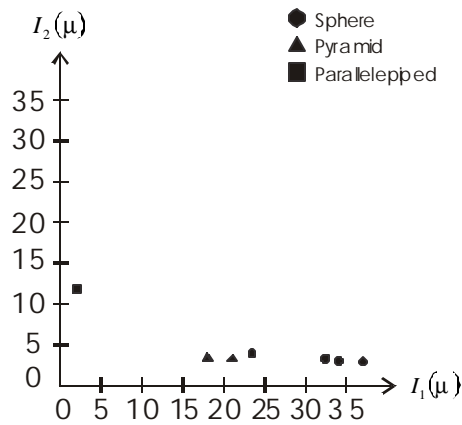


Figure 3. Values of invariant moments I_1 and I_2 for the transformed versions of the objects of Figure 1

Table 1. A comparison of complexity in computing all the moments of order up to $(p+q+r \leq K=3)$ from a discrete image of $N \times N \times N$ voxels.

Method	Multiplications	Additions
Straightforward	$\left(\frac{1}{6}K^3 + K^2 + \frac{11}{6}K + 1\right)N^3$	$\left(\frac{1}{6}K^3 + K^2 + \frac{11}{6}K + 1\right)N^3$
Cygancki <i>et al.</i>	$\left(\frac{1}{2}K^2 - \frac{1}{2}K\right)N^2$	$(K+1)N^3 + \left(\frac{1}{2}K^2 + \frac{7}{2}K + 4\right)N^2$
Li and Shen	0	$\left(\frac{1}{2}K^4 + \frac{1}{2}K^3 + 1\right)(N^3 - 3N^2)$
Li and Ma	$2KN^2$	$N^3 + (K^2 + 2K)N^2$
Yang <i>et al.</i>	$2KN^2$	$\left(\frac{1}{2}K^2 + \frac{7}{2}K + 3\right)N^2$
New	$36t = 36\left(\frac{N}{2} - 1\right)$	$8t = 8\left(\frac{N}{2} - 1\right)$

Table 2. Transformations used to obtain the objects of Figure 2

Transformation N of object M	Translation			Rotation in degrees			Change of scale Scaling factor
	x	y	z	With respect to x	With respect to y	With respect to z	
1,1	20	30	30	20	40	10	2
2,1	15	45	20	30	15	36	3
3,1	37	58	12	40	37	26	1.5
1,2	35	35	21	50	77	16	1.4
2,2	35	34	21	70	44	20	1.7
2,3	18	35	45	10	16	5	3
3,1	30	40	50	40	0	0	0.5
3,2	44	26	12	50	25	16	1.5
3,3	32	12	17	10	77	62	2.5

Table 3. Values of invariant moments $I1$ and $I2$ for the transformed versions of the objects of Figure 1. The times were obtained with a PC Pentium II at 233 MHz.

Transformatiof N of object M	Translation			Rotation in degrees			Change of scale
	x	y	z	With respect to x	With respect to y	With respect to z	Scaling factor
1,1	20	30	30	20	40	10	2
2,1	15	45	20	30	15	36	3
3,1	37	58	12	40	37	26	1.5
1,2	35	35	21	50	77	16	1.4
2,2	35	34	21	70	44	20	1.7
2,3	18	35	45	10	16	5	3
3,1	30	40	50	40	0	0	0.5
3,2	44	26	12	50	25	16	1.5
3,3	32	12	17	10	77	62	2.5

Conclusions and Future Research

A new method to compute geometric moments for a 3D object has been presented. Initially, the object is partitioned in a set of convex balls whose moment evaluation can be reduced to the computation of very simple formulae. The resulting shape moments are finally obtained by addition of the moments of each ball forming the partition, giving that the intersections are empty. Mathematical Morphology was used to obtain the desired partition of the shape. As the morphological operations implied could be computed in a massively parallel computer, the computation of geometric moments is extremely very fast.

One of the main features of the proposed method is that once the partition is obtained its complexity is of $O(N)$.

Derived invariants derived through the standard moments obtained by the proposed methodology can be also used to differentiate among 3-D objects.

As a future work, we pretend to install this technology to sequential algorithms able to

compete with actual algorithms on those computing platforms.

Acknowledgments

The authors would like to thank the CIC-IPN, COFAA, CGPI under projects 20050156 and 20060517, and the CONACYT under project 46805 for their economical support to develop this work.

References

- Albregtsen F., Schulerud H. and Yang L. (1995). Texture Clas si fication of Mouse Liver Cell Nuclei Using Invariant Moments of Consistent Regions, CAIP 95, Proceedings, LNCS 970, pp. 496-502.
- Budrikis Z.L. and Hatamian M.(1984). Moment Calculations by Digital Filters. *AT&T Bell Lab. Tech. J.* 63:217-229.
- Cyganski D., Kreda S.J. and Orr J.A. (1988). Solving for the General Linear Transformation Relating 3-D Objects from the Minimum Moments, In: SPIE Intelligent Robots and Computer Vision VII, Proceedings of the SPIE,

- Vol. 1002, pp. 204-211, Bellingham, WA.
- Dudani S.A., Breeding K.J. and Mcghee R.B. (1977). Aircraft Identification by Moment Invariants. *IEEE Transactions on Computers*, 28(1):39-46.
- Flusser J. and Suk T. (1993). Pattern Recognition by Affine Moment Invariants. *Pattern Recognition*, 26(1): 167-174.
- Flusser J. and Suk T. (1994). A Moment Based Approach to Registration of Images with Affine Distortion. *IEEE Transactions on Geo-science and remote sensing*, 32(2):382-387.
- Flusser J. and Suk T. (1993). Pattern Recognition by Affine Moment Invariants. *Pattern Recognition*, 26(1): 167-174.
- Fu Ch.W., Yen J.Ch. and Chang Sh. (1993). Calculation of Moment Invariants via Hadamard Transform. *Pattern Recognition*, 26(2):287-294.
- Hu M.K. (1962). *Visual Pattern Recognition by Moment Invariants*. IRE Transactions on Information Theory, pp. 179-187.
- Jiang X.Y. and Bunke H. (1991). Simple and Fast Computation of Moments. *Pattern Recognition*, 24(8):801-806.
- Li B.C. (1993). The Moment Calculation of Polyhedra. *Pattern Recognition*, 26: 1229-1233.
- Li B.C. and Shen J. (1992). Pascal Triangle-Transform Approach to the Calculation of 3D Moments. *CVGIP: Graphical Models and Image Processing*, 54:301-307.
- Li B.C. and Ma S.D. (1994). Efficient Computation of 3D Moments, In: Proceedings of 12th the International Conference on Pattern Recognition, Vol. 1, pp. 22-26. Lo C.H. and Don H.S. (1989). 3-D Moment Forms, their Construction and Application to Object Identification and Positioning. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 11:1053-1064.
- Li B.Ch. and Shen J. (1991). Fast Computation of Moment Invariants. *Pattern Recognition*, 24(8):807-813.
- Li B.Ch. (1993). A New Computation of Geometric Moments. *Pattern Recognition*, 26(1):109-113.
- Pei S.C. and Liou L.G. (1994). Using Moments to Acquire the Motion Parameters of a Deformable Object Without Correspondences. *Image Vision Computing*, 12:475-485.
- Philips W. (1993). A New Fast Algorithm for Moment Computation. *Pattern Recognition*, 26(11):1619-1621.
- Reeves A.P., Akey M.L. and Mitchell O.R. (1983). A Moment Based Two-Dimensional Edge Operator, In: Proceedings of the International Conference on Computer Vision and Pattern Recognition, pp. 312-317.
- Reiss T. (1991). The Revisited Fundamental Theorem of Moment Invariants. *IEEE Transactions on Image Analysis and Machine Intelligence*, 13(8): 830-834.
- Sadjadi F.A. and Hall E. (1980). Three-Dimensional Moment Invariants. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. PAMI-2, No 2, 1980.
- Shen J. and Li B.C. (1993). Fast Determination of Center and Radius of Spherical Surface by use of Moments, In: Proceedings of the 8th Scandinavian Conference on Image Analysis, Tromso, Norway, pp. 565-572.
- Sossa H., Mazaira I. and Ibarra J.M. (1999). An Extension to Philips Algorithm for Moment Calculation. *Computación y Sistemas*, 3(1):5-16.
- Sossa H., Yáñez C. and Díaz J.L. (2001). Computing Geometric Moments Using Morphological Erosions. *Pattern Recognition*, 34(2):271-276.
- Yang L., Albrechtsen F., Lonnestad T., Grottum P., Iversen J.G., Rotnes J.S. and Rottingen J.A. (1995). Measuring Shape and Motion of White Blood Cells from a Sequence of Fluorescence Microscopy Images, In: Theory and Applications of Image Analysis II (G.

- Borgefors, Ed), pp. 305-316, World Scientific, Singapore.
- Yang L. and Albrechtsen F. (1994). Fast Computation of Invariant Geometric Moments: a New Method Giving Correct Results. Proceedings of the 12th International Conference on Pattern Recognition, Jerusalem, Israel, 201-204.
- Yang L. and Albrechtsen F. (1996). Fast and Exact Computation of Cartesian Geometric Moments Using Discrete Green's Theorem. *Pattern Recognition*, 29(7):1061-1073.
- Yang L. and Albrechtsen F. and Taxt T. (1997). Fast Computation of Three-Dimensional Geometric Moments Using a Discrete Divergence Theorem and a Generalization to Higher Dimensions. *CGVIP: Graphical models and image processing*, 59(2):97-108.
- Zakaria M.F., Vroomen L.J., Zsombor-Murray P.J.A. and Van-Kessel J.M.H.M. (1987). Fast Algorithm for the Computation of Moment Invariants. *Pattern Recognition*, 20(6):639-643.

Semblanza de los autores

Juan Humberto Sossa-Azuela. Received his BS degree in Communications and Electronics from the University of Guadalajara in 1980. He obtained his Master degree in Electrical Engineering from CINVESTAV-IPN in 1987 and his PhD in Informatics from the INPG, France in 1992. He is currently a titular professor of the Pattern Recognition Laboratory of the Center for Computing Research, Mexico since 1996. He has more than 30 publications in international journals with rigorous refereeing and more than 100 works in national and international conferences. His research areas are Pattern Recognition, Image Analysis and Neural Networks.

Fran cisco Cuevas de la Rosa. Received his BS degree in Computer Sciences from the ITESM, Mexico in 1984. He obtained his Master degree in Computer Science in Artificial Intelligence from the ITESM, Mexico in 1995 and his PhD in Optics Metrology from Centro de Investigaciones en Óptica, A.C., Mexico in 2000. He has more than 20 publications in international journals with rigorous refereeing and more than 40 works in national and international conferences. His research areas are Computer Vision, Optical Metrology, Genetic Algorithms and Neural Networks.

Carlos Aguilar-Ibañez. Received the M. Sc. and Ph. D. in Electrical Engineering from the CINVESTAV-IPN, Mexico in 1994 and 1999, respectively. He is currently a titular professor of the Automatic Control Laboratory of the Center for Computing Research of the, Mexico since 1999. His research interests include nonlinear system, underactuated mechanical system, identification system.

Héctor Benítez-Muñoz. Received his BS degree in Informatics from the Instituto tecnológico de Apizaco, Mexico in 1998. He obtained his Master degree in Computer Science from the Center for Computing Research of the, Mexico in 2005. Actually, he is working at Intermecc Technologies of Mexico where he develops wireless applications for mobile systems.