

# Efficient Computation of Three-dimensional Geometric Moments Based on Object Partition

Humberto Sossa  
Héctor Benítez

Centro de Investigación en Computación,  
Instituto Politécnico Nacional.  
Av. Juan de Dios Bátiz esquina M. Othón de Mendizábal,  
Col. Nueva Industrial Vallejo, México, DF., CP 07738,  
MÉXICO

Tel. 5729 6000 ext. 56512, Fax 5729 6000 ext. 56607

email: hsossa@cic.ipn.mx  
hbenitez75@hotmail.com

Recibido el 29 de octubre de 2004; aceptado el 22 de junio de 2005.

## 1. Abstract

We present a method for computing the three-dimensional moments of an object. The method is based on the idea that the object of interest is first decomposed in a set of cubes under  $d_\infty$ . This decomposition is known to form a partition. The required moments are computed as a sum of the moments of the elements of the partition. The moments of each cube can be calculated in terms of a set of very simple formula using the center of the cube and its radio. The method provides integral accuracy by applying the exact definition of moments. The desired partition is obtained both by morphological erosions and the distance transformation of the image. Both variants are compared, showing that the one using the distance transform is much faster, making it comparable to other traditional sequential approaches. Another interesting feature of the proposed idea to compute the geometric moments of a 3-D object is that once the partition is obtained, moment computation is much faster than earlier methods. Its complexity is in fact of  $O(N)$ .

## 2. Resumen (Cómputo eficiente de momentos geométricos tridimensionales basado en la partición de un objeto)

Se presenta un método para calcular los momentos tridimensionales de un objeto. El método se basa en la idea de

que el objeto se divide primero en un conjunto de cubos bajo  $d_\infty$ , esta descomposición se conoce como partición. Los momentos requeridos se calculan como una suma de momentos de los elementos de la partición. Los momentos de cada cubo pueden calcularse en términos de un conjunto simple de fórmulas utilizando el centro del cubo y su radio. Este método otorga una precisión de tipo integral al aplicar la definición exacta de los momentos. La partición deseada se obtiene tanto por erosiones morfológicas como por la transformación de distancia de una imagen. Ambas variantes se comparan y se muestra que en aquella, donde se utiliza la transformada de distancia, es más rápida, incluso al compararla con otras aproximaciones tradicionales. Otra característica de esta propuesta para calcular los momentos geométricos de un objeto tridimensional es que una vez obtenida la partición, el cálculo de los momentos se realiza de manera más rápida comparada con otros métodos. Su complejidad tiene un factor de  $O(N)$ .

**Key words:** 2-D geometric moments, 3-D geometric moments, mathematical morphology, distance transform.

## 3. Introduction

The two-dimensional moment (for short 2-D moment) of a 2-D object  $R$  is defined as [1]:

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

where  $f(x, y)$  is the characteristic function describing the intensity of  $R$ , and  $p+q$  is the order of the moment. In the discrete case, the double integral is often replaced by a double sum giving as a result:

$$m_{pq} = \sum \sum_R x^p y^q f(x, y) \quad (2)$$

with  $f(x, y)$ ,  $p$  and  $q$  defined in equation (1), where  $(x, y) \in \mathbf{Z}^2$ .

The three-dimensional geometric moment (for short 3-D moment) of order  $p+q+r$  of a 3-D object is defined as [2]:

$$m_{pqr} = \iiint_R x^p y^q z^r f(x, y, z) dx dy dz \quad (3)$$

where  $R$  is a 3-D region. In the discrete case, the triple integral is often replaced by the triple sum giving as a result:

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

with  $f(x, y, z)$ ,  $p, q$  and  $r$  defined in equation (3), where  $(x, y, z) \in \mathbf{Z}^3$ .

In the binary case, the characteristic function takes only values 1 or 0, assuming that for the volume of interest  $f(x, y, z) = 1$ . When we replace this value in equation (4) we get the equation to compute the moments of order  $(p+q+r)$  of a 3-D binary image  $R$  as

$$m_{pqr} = \sum \sum \sum_R x^p y^q z^r \quad (5)$$

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

2-D moments are useful features in the case 2-D shapes. They have been widely used in image analysis. Their applications range from: edge detection [3], texture analysis [4], movement estimation [5], image alignment [6], object description [7], until object recognition [8 and 9]. Due to their usefulness lots of efforts have been done to reduce the time of computation. Among the most important works we can mention the works of Zakaria *et al.* [10], Li and Shen [11], Jiang and Bunke [12], Li [13], Fu *et al.* [14], Philips [15], Yang and Albrechtsen [16 and 17], Flusser [18] and Sossa *et al.* [19 and 27].

The world we live in is generally three-dimensional. 3-D shape information for an object can be obtained by computer tomographic reconstruction, passive 3-D sensors, and active range finders. As 2-D moments, 3-D moments have been used in 3-D image analysis tasks including movement estimation [5], shape estimation [20], and object recognition [2].

Several methods have been proposed to compute the 3-D moments. In [22], 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.* [21], Li and Shen [23] and Li and Ma [24] use a pixel representation of the object. The difference among these methods is the way they compute the moments. Cyganski *et al.*, for example, make use of the filter proposed in [25] while 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 relate 3-D moments with the so-called LT moments that are easier to evaluate. Although these methods reduce

the number of operations to compute the moments, they require a computation of  $O(N^3)$ , where  $N$  is the size of the object in one of the dimensions. Recently, Yang, Albrechtsen and Taxt [26] proposed to use the so-called discrete divergence theorem to compute the 3D moments of an object. They got a reduction in the number of operations to  $O(N^2)$ .

In this note we present a different approach to efficiently compute the 3-D moments of a binary object in  $\mathbf{Z}^3$ , with complexity of  $O(N)$ . It is based on the following idea [27]:

1. Decompose the given object into a union of disjoint cubes.
2. Compute the geometric moments for each of these cubes, and
3. Obtain the final moments as a sum of the moments computed for each cube.

### 3.1 Outline of the paper

The remaining of the paper is organized as follows. In the second section the proposed formulae to compute the geometric moments of a 3-D object are given. They are obtained by means of equation (3) and yielding thus accurate results. In Section 3 a short comparison between the formulae obtained in terms of equations (3) and (5) is given. In section 4 we describe two ways to segment the object into a set of cubes. The first method uses morphological erosions. The second one uses the three-dimensional distance transform of the image. In Section 5 several experiments are given showing that if the distance transform is used to obtain the desired partition, smaller processing times to compute the geometric moments of a 3-D object are obtained. In Section 6 the complexity of the computation once the partition is obtained is given, showing that this complexity of order  $O(N)$ . In Section 7 we present a short comparison between the proposal and the conventional method since the precision point of view. Section 8 finally provides conclusions of the work.

## 4. Development

### 4.1. Moments of a cube

In the last section we mentioned that to compute the desired moments of a 3-D object, it should be first decomposed into a set of cubes. We also said that a set of simple expressions should then be applied to get the desired values. In this section, this set of expressions is derived.

Before deriving the set of expressions, it is important to know that depending on the definition of moments used, the set of

expressions obtained might differ resulting in some differences. This situation was first studied in [28] and recently re-discussed in [29], both in the 2-D case. As stated in [29], if  $M_{pq}$  are the 2-D moments obtained by means of equation (1) and  $m_{pq}$  those obtained in terms of equation (2) a difference  $|M_{pq} - m_{pq}|$  is introduced due to the approximations and numeric integration of  $x^p y^q$  over each pixel of  $R$ .

In both papers [28] and [29] the authors prove that the moments obtained in terms of equation (1) yield exact results. Following the same idea we propose to use equation (3) instead of equation (5) to derive the set of necessary equations to get also more accurate results when computing the 3-D geometric moments for a 3-D object.

#### 4.2 Set of expressions

To derive the set of expressions needed to accurately compute the desired 3-D moments, let us consider a cube centered in  $(X_c, Y_c, Z_c)$ , 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  $(X_c + t, Y_c + t, Z_c + t)$ .

The characteristic function of this block is

$$f(x, y, z) = \begin{cases} 1 & \text{if } (x, y, z) \in (a, b) \times (c, d) \times (e, f) \\ 0 & \text{otherwise} \end{cases}$$

with

$$\begin{aligned} a &= X_c - t - 0.5 \\ b &= X_c + t + 0.5 \\ c &= Y_c - t - 0.5 \\ d &= Y_c + t + 0.5 \\ e &= Z_c - t - 0.5 \\ f &= Z_c + t + 0.5 \end{aligned}$$

According to equation (3), the exact moments of a cube are given as

$$\begin{aligned} M_{pqr} &= \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} x^p y^q z^r f(x, y, z) dx dy dz \\ &= \frac{1}{(p+1)(q+1)(r+1)} [b^{p+1} - a^{p+1}] \cdot [d^{q+1} - c^{q+1}] \cdot [f^{r+1} - e^{r+1}] \end{aligned} \tag{6}$$

The reader can easily verify that the first 20 expressions for the moments are:

$$M_{000} = (2t + 1)^3 \tag{7a}$$

$$M_{100} = M_{000} X_c \tag{7b}$$

$$M_{010} = M_{000} Y_c \tag{7c}$$

$$M_{001} = M_{000} Z_c \tag{7d}$$

$$M_{200} = \frac{M_{000}}{3} (3X_c^2 + t(t+1) + 0.25) \tag{7e}$$

$$M_{020} = \frac{M_{000}}{3} (3Y_c^2 + t(t+1) + 0.25) \tag{7f}$$

$$M_{002} = \frac{M_{000}}{3} (3Z_c^2 + t(t+1) + 0.25) \tag{7g}$$

$$M_{110} = M_{100} Y_c = M_{000} X_c Y_c \tag{7h}$$

$$M_{101} = M_{100} Z_c = M_{000} X_c Z_c \tag{7i}$$

$$M_{011} = M_{001} Y_c = M_{000} Y_c Z_c \tag{7j}$$

$$M_{300} = M_{000} X_c (X_c^2 + t(t+1) + 0.25) \tag{7k}$$

$$M_{030} = M_{000} Y_c (Y_c^2 + t(t+1) + 0.25) \tag{7l}$$

$$M_{003} = M_{000} Z_c (Z_c^2 + t(t+1) + 0.25) \tag{7m}$$

$$M_{120} = \frac{M_{000}}{3} X_c (3Y_c^2 + t(t+1) + 0.25) \tag{7n}$$

$$M_{210} = \frac{M_{000}}{3} Y_c (3X_c^2 + t(t+1) + 0.25) \tag{7o}$$

$$M_{102} = \frac{M_{000}}{3} X_c (3Z_c^2 + t(t+1) + 0.25) \tag{7p}$$

$$M_{201} = \frac{M_{000}}{3} Z_c (3X_c^2 + t(t+1) + 0.25) \tag{7q}$$

$$M_{012} = \frac{M_{000}}{3} Y_c (3Z_c^2 + t(t+1) + 0.25) \tag{7r}$$

$$M_{021} = \frac{M_{000}}{3} Z_c (3Y_c^2 + t(t+1) + 0.25) \tag{7s}$$

$$M_{111} = M_{000} X_c Y_c Z_c \tag{7t}$$

In [32] the authors have proved that the same set of 20 expressions obtained through equation (5) is the following:

$$m_{000} = (2t + 1)^3 \tag{8a}$$

$$m_{100} = m_{000} X_c \tag{8b}$$

$$m_{010} = m_{000} Y_c \tag{8c}$$

$$m_{001} = m_{000} Z_c \tag{8d}$$

$$m_{200} = \frac{m_{000}}{3} (3X_c^2 + t(t+1)) \quad (8e)$$

$$m_{020} = \frac{m_{000}}{3} (3Y_c^2 + t(t+1)) \quad (8f)$$

$$m_{002} = \frac{m_{000}}{3} (3Z_c^2 + t(t+1)) \quad (8g)$$

$$m_{110} = m_{100} Y_c = m_{000} X_c Y_c \quad (8h)$$

$$m_{101} = m_{100} Z_c = m_{000} X_c Z_c \quad (8i)$$

$$m_{011} = m_{001} Y_c = m_{000} Y_c Z_c \quad (8j)$$

$$m_{300} = m_{000} X_c (X_c^2 + t(t+1)) \quad (8k)$$

$$m_{030} = m_{000} Y_c (Y_c^2 + t(t+1)) \quad (8l)$$

$$m_{003} = m_{000} Z_c (Z_c^2 + t(t+1)) \quad (8m)$$

$$m_{120} = \frac{m_{000}}{3} X_c (3Y_c^2 + t(t+1)) \quad (8n)$$

$$m_{210} = \frac{m_{000}}{3} Y_c (3X_c^2 + t(t+1)) \quad (8o)$$

$$m_{102} = \frac{m_{000}}{3} X_c (3Z_c^2 + t(t+1)) \quad (8p)$$

$$m_{201} = \frac{m_{000}}{3} Z_c (3X_c^2 + t(t+1)) \quad (8q)$$

$$m_{012} = \frac{m_{000}}{3} Y_c (3Z_c^2 + t(t+1)) \quad (8r)$$

$$m_{021} = \frac{m_{000}}{3} Z_c (3Y_c^2 + t(t+1)) \quad (8s)$$

$$m_{111} = m_{000} X_c Y_c Z_c \quad (8t)$$

We can use equation (7) to derive higher order geometric moments. We have found however the following equivalent expression, in terms of moment  $M_{000}$ , more suitable for this job:

$$M_{pqr} = \frac{M_{000}}{(p+1)(q+1)(r+1)} PQR \quad (9)$$

for  $p$  even:

$$P = (p+1)X_c^p + \binom{p+1}{3} X_c^{p-1} h^2 + \dots + h^p$$

for  $p$  odd:

$$P = (p+1)X_c^p + \binom{p+1}{3} X_c^{p-1} h^2 + \dots + \binom{p+1}{p} X_c h^{p-1}$$

for  $q$  even:

$$Q = (q+1)Y_c^q + \binom{q+1}{3} Y_c^{q-1} h^2 + \dots + h^q$$

for  $q$  odd:

$$Q = (q+1)Y_c^q + \binom{q+1}{3} Y_c^{q-1} h^2 + \dots + \binom{q+1}{q} Y_c h^{q-1}$$

for  $r$  even:

$$R = (r+1)Z_c^r + \binom{r+1}{3} Z_c^{r-1} h^2 + \dots + \binom{r+1}{r} Z_c h^{r-1}$$

for  $r$  odd:

$$R = (r+1)Z_c^r + \binom{r+1}{3} Z_c^{r-1} h^2 + \dots + \binom{r+1}{r} Z_c h^{r-1}$$

where  $h = t + 0.5$

and

$$\binom{n}{k} = \frac{n!}{k!(n-k)!}$$

Some examples are shown below:

$$\begin{aligned} M_{211} &= \frac{M_{000}}{(3)(2)(2)} [3X_c^2 + h^2] 2Y_c [2Z_c] \\ &= \frac{M_{000}}{3} Y_c Z_c [3X_c^2 + t(t+1) + 0.25] \end{aligned}$$

$$\begin{aligned} M_{423} &= \frac{M_{000}}{(5)(3)(4)} [5X_c^4 + 10X_c^2 h^2 + h^4] [3Y_c^2 + h^2] [4Z_c^3 + 4Z_c h^3] \\ &= \frac{M_{000}}{15} Z_c [5X_c^4 + 10X_c^2 h^2 + h^4] [3Y_c^2 + h^2] [Z_c^2 + h^3] \end{aligned}$$

### 4.3 Comparison

While equation (3) yields exact results, equation (5) provides some moments with small errors due to the zero-order approximation for numerical integration when using sums. We will always find  $M_{pqr} \geq m_{pqr}$ . The error  $M_{pqr} - m_{pqr}$  depends directly on  $p$ ,  $q$  and  $r$ . You can easily verify that:

$$M_{200} - m_{200} = \frac{m_{000}}{12}$$

$$M_{020} - m_{020} = \frac{m_{000}}{12}$$

$$M_{002} - m_{002} = \frac{m_{000}}{12}$$

$$M_{300} - m_{300} = \frac{m_{100}}{4}$$

$$M_{030} - m_{030} = \frac{m_{100}}{4}$$

$$M_{003} - m_{003} = \frac{m_{100}}{4}$$

On the other hand for some moments no errors are produced:

$$M_{000} - m_{000} = 0$$

$$M_{100} - m_{100} = 0$$

$$M_{010} - m_{010} = 0$$

$$M_{001} - m_{001} = 0$$

$$M_{110} - m_{110} = 0$$

$$M_{011} - m_{011} = 0$$

$$M_{111} - m_{111} = 0$$

#### 4.4 Obtaining the partition

At the beginning of this note we mentioned that to compute the desired moments of a 3-D object, it should be first decomposed into a set of cubes. It is not difficult to see that this step is the most critical one to get small processing times. In this section we describe two methods to partition an object according to the idea introduced at the end of Section I. The first one emerges as a natural extension of the proposal described in [27], useful to get the geometric moments of a 2-D object. The second proposal uses the distance transform of the image to get the necessary information of each cube, being a generalization of the method described in [31].

##### 4.4.1 Decomposition by morphological operations

The following method to compute the geometric moments of a 3-D object  $R \subset \mathbf{Z}^3$ , using morphological erosions is an extension of the one described in [27]. It is composed of the following steps:

1. Initialize as many accumulators as needed  $C_i=0$ , for  $i = 1, 2, \dots, M$ , one for each geometric moment.
2. Make  $A = R$  and  $B = \{(\pm a, \pm b, \pm c) \mid a, b, c \in \{-1, 0, 1\}\}$ ,  $B$  is a  $3 \times 3 \times 3$  pixel neighborhood in  $\mathbf{Z}^3$ .
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.

4. Select one of the points of  $A$  and given that the radius  $t$  of the maximal cube is known, we use the formulae derived in the last section to compute the moments of this maximal cube, the resulting values are added to the respective accumulator,  $C_i$ , for  $1, 2, 3, \dots, M$ .
5. Eliminate this ball from region  $R$ , and assign this new set to  $R$ .
6. Repeat steps 2 to 5 with the new  $R$  until it becomes  $\emptyset$ .

The method just described gives us as a result the values of the geometric moments of order  $(p+q+r) \leq 3$ , using only erosions and the formulae developed in Section 4.2.

By their nature, the erosions can be executed 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 either by equation (7) or equation (8), to compute the geometric moments of these maximal cubes, and add these values to the respective accumulators.
2. Eliminate the maximal cubes from region  $R$ , and assign this new set to  $R$ .

The enhancement (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.

##### 4.4.2 Decomposition by distance transform

The following procedure (an extension of the one proposed in [31]) decomposes the 3-D object  $R \subset \mathbf{Z}^3$ , onto cubes by means of the well-known distance transform [30] that provides with all the necessary information of all the maximal cubes that cover the volume of the object:

1. Initialize as many accumulators as needed  $C_i=0$ , for  $i=1, 2, \dots, M$ , one for each geometric moment.

2. Obtain the distance transform  $DT(p)$  for each point  $p$  of region  $R$  and at the same time  $m = \max\{DT(x) \in R\}$ .
3. Following an ordered sweeping, look for a  $p \in R$ , such as  $DT(p) = m$ ,  $DT(p)$  is the radius of the open cube, such as the radius of the *closed cube* under question is  $DT(p) - 1$ .
4. Compute the moments for the corresponding *closed cube* for this value, updating the corresponding accumulators  $C_p$  for  $1, 2, 3, \dots, M$  using the formulae given equation (7) or (8).
5. Eliminate the *closed cube* of region  $R$ , and assign this new set to  $R$ .
6. Following the same sweeping, look for a new  $p$  for which  $DT(p) = m$  and that do not intersects with the previously selected cubes during the sweeping. Execute steps 4 and 5. Repeat this step until the sweeping ends, being sure that all maximal cubes are eliminated.
7. Repeat steps 2 to 6 until  $R$  becomes  $\emptyset$ .

As in the case of method described in Section 4.4.1, the method just described yields the true values of the geometric moments of order  $(p+q+r) \leq 3$ , using the distance transform of the image and the formulae developed in Section 4.2.

As it is known by its nature, the distance transform can be implemented on a massively parallel computer in pretty short processing times. However, the method proposed in [30] to compute the distance transform of an image is also sequential that provides, as we will see in Section 4.5, very good processing times on a conventional platform.

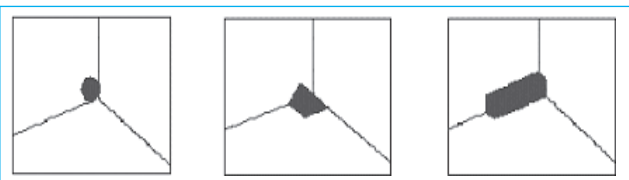
#### 4.5 Experimental results

In this section the iterative morphological (IM), enhanced morphological (EM) and distance transform (DT) approaches useful to obtain the desired partition to get the geometric moments of a 3-D object and described in Section 4.4 are compared.

##### 4.5.1 Results with random generated synthetic images

In this first experiment, the three methods were tested on several dozens of synthetically generated images, all of them 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 set zero.

The IM method takes on average, over the whole set of images, 255 seconds to compute all moments of order  $(p+q+r) \leq 3$  on a



**Fig. 1.** Other objects used to test the performance of the three proposed methods to compute the desired geometric moments of a 3-D object. Object No.1 (Sphere of radio 8), Object No.2 (Pyramid with base of 25) and Object No.3 (Parallelepiped of  $55 \times 13 \times 13$ ).

550 MHz based system. The 255 seconds include the time to compute the partition iteration by iteration. The EM method requires only about 65 seconds on the same platform compute the same moments. Again the 50 seconds include the time to get the partition. The DT method is the fastest of the three proposed approaches. It takes on average 12 seconds on the same platform to get the moments. In the three cases most of the time is required to obtain the necessary partitions.

##### 4.5.2 Results with manually generated images

The IM, EM and DT method were also tested with images of more realistic objects, a sphere, a pyramid and parallelepiped in this case (Figure 1). The three objects were translated, rotated and scaled to get several transformed versions of them. Three of these transformed versions are shown in Figure 2.

Table 1 shows the time invested by the three proposals (IM, EM and DT) to compute the 20 moments on the three objects. As you can appreciate all three approaches perform very fast for the parallelepiped and but very slow for the sphere.

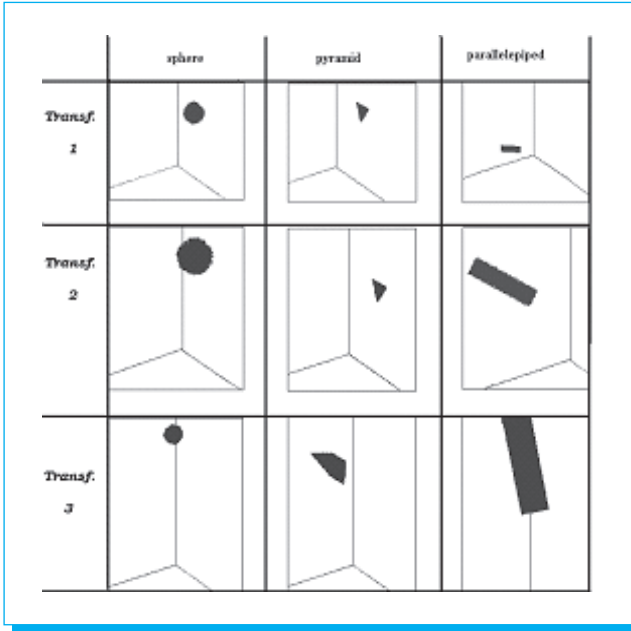
From this we can immediately see that the distance transform based proposal is the fast of all proposals.

##### 4.6 Efficiency and computation

With respect to other methods our proposal is faster, once the partition is obtained. As you can appreciate its complexity is

**Table 1.** Average times in seconds provides by the three proposals for the objects used in the experiments.

Object	Average time in seconds BMF approach	Average time in seconds EM approach	Average time in seconds DT approach
Sphere	242.809	69.374	19.821
Pyramid	197.822	43.004	11.317
Parallelepiped	25.192	6.238	1.612



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

of  $O(N)$ . To show this, and without loss of generality, 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  $N \times N \times N$  voxels, with  $t$  its radio.

The number of operations required by one of the fastest methods (for example the method of Yang, Albrechtsen and Takt, [26]) to compute all the moments of order  $(p+q+r)$  up to some  $K$ , let say  $K=3$  from a discrete image of  $N \times N \times N$  pixels is:  $2KN^2$  multiplications and  $\left(\frac{1}{2}K^2 + \frac{7}{2}K + 3\right)N^2$  additions (for the details, refer to [26]).

The number of operations required by our proposal, *once the partition has been obtained*, will depend basically on the radius  $t$  of the object:

$$36t = 36\left(\frac{N}{2} - 1\right) \text{ multiplications}$$

$$\text{and } 8t = 8\left(\frac{N}{2} - 1\right) \text{ additions.}$$

To get these two numbers we just added the number of multiplications and additions required by each of the 20

**Table 2.** Relative differences between moments.

	$e_1$	$e_2$	$e_3$
<b>Sphere</b>	0.0064	0.0063	0.0063
<b>Pyramid</b>	0.0029	0.0091	0.0014
<b>Parallelepiped</b>	0.0003	0.0058	0.0058

moments to be computed.  $m_{000}$  requires, for example, three multiplications and one addition.  $m_{100} = m_{000} X_c$  requires one multiplication and no additions because it is supposed that the term  $(2t + 1)^3$  was already computed.

The careful reader can rapidly see that our method is faster than any other, 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.

#### 4.7 Impact on the value of the moments

In this section we present a short analysis of the effect of using either the expressions given by equation (7) or (8). For this we make use again the three objects shown in figure 1. Table 2 shows the relative errors of central moments:  $M_{200}$ ,  $M_{020}$ ,  $M_{002}$ ,  $\mu_{200}$ ,  $\mu_{020}$ , and  $\mu_{002}$ . Relative errors that were investigated were:

$$e_1 = \frac{M_{200} - \mu_{200}}{M_{200}}, e_2 = \frac{M_{020} - \mu_{020}}{M_{020}} \text{ and } e_3 = \frac{M_{002} - \mu_{002}}{M_{002}}.$$

Central moments  $\mu_{pqr}$  were obtained as

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

Central moments  $M_{pqr}$  were obtained by means of the expressions given by equation (7). Errors  $e_1$ ,  $e_2$  and  $e_3$  correspond to each one of the three axes:  $x$ ,  $y$  and  $z$ . From Table 2, we can conclude that for elongated objects (where the prolongation is parallel to one of the axes), the relative error is much smaller (0.0003 for the parallelepiped in the  $x$  axis). In this case, as can be appreciated for compact objects (as the sphere), the errors are bigger. As a result, in the general case, it is better to use the expressions given by equation (7) for the computation of the geometric moments, with certainly will produce a better description and in consequence an accurate recognition of the objects. The bigger errors for the spherical object can be due also to digitalization problems.

## 5. Conclusion

In this note we have presented a very simple method to accurately compute the geometric moments for a 3-D object. Initially, the object is partitioned into a set of convex cubes whose moment evaluation can be reduced to the computation of very simple formulae.

These expressions were derived from the original definition of moments given by equation (3). This gives more accurate values for the moments. This would not happen if equation (5) would be used. An error is introduced due to zero-order approximation and numeric integration of  $x^p y^q z^r$  over each pixel. The resulting shape moments are finally obtained by addition of the moments of each ball forming the partition, giving that the intersections are empty.

To obtain the desired object partition we have used to variants, one based on morphological erosions, the other based on the distance transform of the image containing the object. The results show that the variant using the distance transform is much more superior. This makes the proposal competitive with traditional methods working on a sequential platform.

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

## Acknowledgments

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

## 6. References

- [1] M. K., Hu, Visual pattern recognition by moment invariants, *IRE Transactions on Information Theory*, 179-187, 1962.
- [2] C. H. Lo and H. S. Don, 3-D moment forms. Their construction and application to object identification and positioning, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 11:1053-1064, 1989.
- [3] A. P. Reeves, M. L. Akey and O. R. Mitchell, A moment based two-dimensional edge operator, in *Proceedings of the International Conference on Computer Vision and Pattern Recognition*, pp. 312-317, 1983.
- [4] F. Albreghsen, H. Schulerud and L. Yang, Texture classification of mouse liver cell nuclei using invariant moments of consistent regions, *CAIP 95, Proceedings, LNCS 970*, pp. 496-502, 1995.
- [5] S. C. Pei and L. G. Liou, Using moments to acquire the motion parameters of a deformable object without correspondences, *Image Vision Computing*, 12:475-485, 1994.
- [6] J. Flusser and T. Suk, A moment based approach to registration of images with affine distortion, *IEEE Transactions on Geo-science and remote sensing*, 32(2):382-387, 1994.
- [7] L. Yang, F. Albreghsen, T. Lonnestad, P. Grottum, J. G. Iversen, J. S. Rotnes and J. A. Rottingen, 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, 1995.
- [8] S. A. Dudani, K. J. Breeding and R. B. Mcghee, Aircraft identification by moment invariants, *IEEE Transactions on Computers*, 28(1):39-46, 1977.
- [9] J. Flusser and T. Suk, Pattern recognition by affine moment invariants, *Pattern Recognition*, 26(1):167-174, 1993.
- [10] M. F. Zakaria, L. J. Vroomen, P. J. A. Zsomrob-Murray and J. M. H. M. van Kessel, Fast algorithm for the computation of moment invariants, *Pattern Recognition*, 20(6):639-643, 1987.
- [11] B. Ch. Li and J. Shen, Fast computation of moment invariants, *Pattern Recognition*, 24(8):807-813, 1991.
- [12] X. Y. Jiang and H. Bunke, Simple and fast computation of moments, *Pattern Recognition*, 24(8):801-806, 1991.
- [13] B. Ch. Li, A new computation of geometric moments, *Pattern Recognition*, 26(1):109-113, 1993.
- [14] Ch. W. Fu, J. Ch. Yen and Sh. Chang, Calculation of moment invariants via Hadamard transform, *Pattern Recognition*, 26(2):287-294, 1993.
- [15] W. Philips, A new fast algorithm for moment computation, *Pattern Recognition*, 26(11):1619-1621, 1993.
- [16] L. Yang and F. Albreghsen, 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, 1994.
- [17] L. Yang and F. Albreghsen, Fast and exact computation of Cartesian geometric moments using discrete Green's theorem, *Pattern Recognition*, 29(7):1061-1073, 1996.
- [18] J. Flusser, Fast calculation of geometric moments of binary images, in *Pattern Recognition and Medical Computer Vision*, M. Gengler (Ed.), pp. 265-274, 1998.
- [19] H. Sossa, I. Mazaira and J. M. Ibarra, An extension to Philips algorithm for moment calculation, *Computación y Sistemas*, 3(1):5-16, 1999.
- [20] J. Shen and B. C. Li, 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, 1993.
- [21] D. Cyganski, S. J. Kreda and J. A. Orr, 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, 1988.



- [22] B. C. Li, The moment calculation of polyhedra, *Pattern Recognition*, 26:1229-1233, 1993.
- [23] B. C. Li and J. Shen, Pascal triangle-transform approach to the calculation of 3D moments, *CVGIP: Graphical Models and Image Processing*, 54:301-307, 1992.
- [24] B. C. Li and S. D. Ma, Efficient computation of 3D moments, in Proceedings of 12 the International Conference on *Pattern Recognition*, Vol 1, pp. 22-26, 1994.
- [25] Z. L. Budrikis and M. Hatamian, Moment calculations by digital filters, *AT&T Bell Lab. Tech. J.* 63:217-229, 1984.
- [26] L. Yang and F. Albrechtsen and T. Taxt, 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, 1997.
- [27] H. Sossa, J. L. Díaz de León and C. Yáñez, Computing geometric moments using morphological erosions, *Pattern Recognition*, 34(2):271-276, 2001.
- [28] M. Dai, P. Baylou and M. Najim, An efficient algorithm for computation of shape moments from run-length codes or chain codes, *Pattern Recognition*, 25(10):1119-1128, 1992.
- [29] J. Flusser, Refined moment calculation using image block representation, *IEEE Transactions on image processing*, 9(11):1977-1978, 2000.
- [30] J. L. Díaz de León and H. Sossa, Mathematical Morphology based on linear combined metric spaces on  $Z^2$  (Part I): Fast distance transforms, *Journal of Mathematical Imaging and Vision*, 12:137-154, 2000.
- [31] J. L. Díaz de León and C. Yáñez, Computing of geometrical moments based on distance transform (In Spanish), *III Taller Iberoamericano de Reconocimiento de Patrones*, Ciudad de México, Marzo 23-27, 1998, pp. 185-195.
- [32] H. Sossa, F. Cuevas, C. Aguilar and H. Benítez, 3-D Cartesian Geometric Moment Computation using Morphological Erosions Applied to Object Recognition, Sent to *Revista Ingeniería, Investigación y Tecnología*.

## ESIME Zacatenco

La Coordinación del Piso de Pruebas  
para Transformadores de los  
Laboratorios Pesados II de Ingeniería Eléctrica  
cuenta con pruebas acreditadas  
que se encuentran al servicio de la industria nacional.

RESISTENCIA DE AISLAMIENTO

RESISTENCIA ÓHMICA

RIGIDEZ DIELECTRICA

TENSIÓN APLICADA

TENSIÓN INDUCIDA

ELEVACIÓN DE TEMPERATURA

IMPULSO

Tel. 5729 6000, ext. 54750  
correo electrónico: piso\_pruebas@hotmail.com