Fast Computation of 3-D Geometric Moments Using a Discrete Gauss' Theorem

Luren Yang, Fritz Albregtsen and Torfinn Taxt

Image Processing Laboratory, Department of Informatics, University of Oslo
P. O. Box 1080 Blindern, N-0316 Oslo, Norway

Abstract. A discrete Gauss' theorem is presented. Using a fast surface tracking algorithm and the discrete Gauss' theorem, we design a new method to compute the Cartesian geometric moments of 3-D objects. Compared to previous methods to compute such moments, the new method reduces the computational complexity significantly.

1 Introduction

In a binary 3-D voxel image, the $(p+q+r)$'th order Cartesian geometric moment of a 3-D object is defined as

$$m_{pqr} = \sum\sum_{R}\sum x^p y^q z^r \tag{1}$$

which is a sum of monomials over a discrete 3-D region R. The 3-D moments have been successfully used in 3-D image analysis [1, 7, 8]. But the application is limited due to the computational complexity. To calculate all the moments with order $0 \leq p+q+r \leq K$, a straightforward method needs additions and multiplications of $O(K^3N^3)$. (Assume that the object is represented by an $N \times N \times N$ voxel image). Some fast computation methods have been proposed [1, 4, 6]. These methods have reduced the number of multiplications and additions, but still require computation of $O(N^3)$.

In this paper, we introduce a discrete Gauss' theorem. Corresponding to Gauss' original theorem for continuous functions, the discrete Gauss' theorem is used to compute a sum of a function over a 3-D discrete region by a summation over the enclosing, discrete surface. Using surface tracking [2] and the discrete Gauss' theorem, we design a new method to compute the 3-D moments of 3-D objects. The new method reduces the computational complexity significantly, requiring computation of $O(N^2)$.

2 Discrete Gauss' Theorem

The original Gauss' theorem relates a surface integral over a closed surface in 3-D to a triple integral over the enclosed region. To deal with the problems in digital image analysis, we introduce a discrete version of Gauss' theorem, which relates a sum of a function over a discrete, closed surface to a sum of the function over the enclosed discrete region. The discrete Gauss' theorem is necessary since applying the original Gauss' theorem directly to digital images may result in considerable errors [9, 10].

First we define some concepts: An object region R in a discrete 3-D space is a finite subset of $\mathbb{Z}^3$. The complement of a 3-D object is called background. Assume that each element in the discrete 3-D space is a cubic voxel with six faces. A voxel face is called a background-to-object transition if the face is shared by a background and an object voxel. The discrete boundary (or the surface) T of an object R is all the background-to-object transitions enclosing the object R. A background-to-object transition is associated with a position (x, y, z) and a direction $(\Delta x,\ \Delta y,\ \Delta z)$. Let (x_1, y_1, z_1) be the background and (x_2, y_2, z_2) be the object voxel corresponding to a transition. We define that $(x, y, z) = (\min(x_1, x_2), \min(y_1, y_2), \min(z_1, z_2))$, $\Delta x = x_2 - x_1$, $\Delta y = y_2 - y_1$ and $\Delta z = z_2 - z_1$. The direction defined in this way is normal to the surface and points towards the object. The discrete Gauss's theorem is then:

Discrete Gauss' Theorem *Let $f(x, y, z)$ be a real function defined on $\mathbb{Z}^3$ that has nonzero values only on a discrete 3-D region R. Let T be the set of all the background-to-object transitions enclosing the region R. Then we have*

$$\sum\sum_R\sum f(x, y, z) = -\sum\sum_T F_x(x, y, z)\Delta x \tag{2}$$

$$\sum\sum_R\sum f(x, y, z) = -\sum\sum_T F_y(x, y, z)\Delta y \tag{3}$$

$$\sum\sum_R\sum f(x, y, z) = -\sum\sum_T F_z(x, y, z)\Delta z \tag{4}$$

where F_x is defined by

$$F_x(x, y, z) = \sum_{i=x_0}^{x} f(i, y, z) \tag{5}$$

in which x_0 is an integer satisfying $(x, y, z) \in R \Rightarrow x_0 < x$. F_y and F_z are defined in the same way.

Proof. Assuming that the object R consists of line segments parallel to the x-axis, and that the x-coordinates of the start point and the end point of the line segment l are $x_1(l)$ and $x_2(l)$. Eq. (2) is proved as follows:

$$\begin{aligned}\sum\sum_R\sum f(x, y, z) &= \sum\sum_l \sum_{x=x_1(l)}^{x_2(l)} f(x, y, z) \\ &= \sum\sum_l \left(F_x(x_2(l), y, z) - F_x(x_1(l) - 1, y, z)\right) \\ &= -\sum\sum_T F_x(x, y, z)\Delta x\end{aligned}$$

Eqs. (3) and (4) can be proved similarly. □

The discrete Gauss's theorem shows that a triple sum may be reduced to a double sum if F_x, F_y or F_z can be expressed in a closed form. However, to reduce computational complexity from $O(N^3)$ to $O(N^2)$, one must have an $O(N^2)$ algorithm to detect the boundary of a 3-D region. Fortunately, such fast surface tracking algorithms exist. The algorithm proposed by Gordon and Udupa [2] is one of the most efficient surface tracking algorithms reported in the literature. Given an array of voxels and one of the background-to-object transitions, the algorithm produces a list of all the transitions which are connected to the given transition. On the average, one-third of the transitions will be visited twice, and the rest once.

3 3-D Moment Computation

Using the discrete Gauss's theorem, we design a new method to compute the geometric moments of 3-D objects represented by voxel images. Substituting $f(x,y,z) = x^p y^q z^r$ into Eq. (2) and setting $x_0 = 0$, we have

$$m_{pqr} = \sum\sum_{T} \sum_{i=0}^{x} i^p y^q z^r \Delta x \tag{6}$$

Rewriting the partial sum of the power series, we obtain

$$m_{pqr} = \sum\sum_{T} \left(\frac{x^{p+1} y^q z^r}{p+1} + \frac{x^p y^q z^r}{2} + \sum_{j=2}^{p} \frac{1}{j} C_p^{j-1} B_j x^{p-j+1} y^q z^r \right) \Delta x \tag{7}$$

where C_p^{j-1} is a binomial coefficient defined by $C_p^s = p!/s!(p-s)!$, and B_j is the j'th Bernoulli number. Let

$$u_{pqr} = \sum\sum_{T} x^{p+1} y^q z^r \Delta x \tag{8}$$

Then Eq. (7) becomes

$$m_{pqr} = \frac{u_{pqr}}{p+1} + \frac{u_{p-1,q,r}}{2} + \sum_{j=2}^{p} \frac{1}{j} C_p^{j-1} B_j u_{p-j,q,r} \tag{9}$$

This is our formula for 3-D moment computation, obtained from Eq. (2) of the discrete Gauss's theorem. Similar formulas can be derived from Eqs. (3) and (4). We see that the moment m_{pqr} can be computed by $O(N^2)$ operations with this formula, in which m_{pqr} is a linear combination of u_{jqr} for $j = 0, \cdots, p$ and u_{jqr} is a sum over the boundary T. The moments of order up to three, which are often used in applications, can be computed as

$$\begin{bmatrix} m_{0qr} \\ m_{1qr} \\ m_{2qr} \\ m_{3qr} \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 1/2 & 1/2 & 0 & 0 \\ 1/6 & 1/2 & 1/3 & 0 \\ 0 & 1/4 & 1/2 & 1/4 \end{bmatrix} \begin{bmatrix} u_{0qr} \\ u_{1qr} \\ u_{2qr} \\ u_{3qr} \end{bmatrix} \tag{10}$$

To calculate u_{pqr} with $0 \leq p+q+r \leq K$, a straightforward method needs additions and multiplications of $O(K^3 N^2)$. We propose a more efficient method to compute this value. Recall that a background-to-object transition is associated with a position coordinate (x, y, z). For a given pair of (y, z), we have a set of transitions. We denote the set by $T(y, z)$ and define $v_p(y, z)$ as

$$v_p(y, z) = \sum_{T(y,z)} x^{p+1} \Delta x \tag{11}$$

Comparing Eq. (8) with Eq. (11), we have

$$u_{pqr} = \sum_y \sum_z v_p(y, z) y^q z^r \tag{12}$$

which means that u_{pqr} is the $(q+r)$'th order moment of a 2-D gray level image with intensity function $v_p(y, z)$.

Hatamian [3] proposed a method for fast computation of the geometric moments of a 2-D gray level image. This method uses a digital filter which we call the Hatamian filter. To compute the moments of a 2-D image, we first filter the image in one direction and then the other direction. Letting $v_p^0(y, z)$ be the result of the Hatamian filtering of $v_p(y, z)$ in the y-direction, we have

$$v_p^0(y, z) = \sum_{k=y}^{N} v_p(k, z) \tag{13}$$

Applying the Hatamian filter recursively in the y-direction, we obtain

$$v_p^q(y, z) = \sum_{k=y}^{N} v_p^{q-1}(k, z) \tag{14}$$

Then we filter the image $v_p^q(y, z)$ in the z-direction, and obtain

$$v_p^{q0}(y, z) = \sum_{k=z}^{N} v_p^q(y, k) \tag{15}$$

Applying the filter recursively in the z-direction gives

$$v_p^{qr}(y, z) = \sum_{k=z}^{N} v_p^{q,r-1}(y, k) \tag{16}$$

The 2-D moments u_{pqr} are then obtained as a linear combination of $v_p^{ij}(1, 1)$ for $i = 0, \cdots, q$ and $j = 0, \cdots, r$. In vector notation, we have

$$\mathbf{u}_p = \mathbf{M}\mathbf{v}_p \tag{17}$$

For $\mathbf{u}_p = [u_{p00}, u_{p01}, u_{p02}, u_{p03}, u_{p10}, \cdots, u_{p33}]^T$ and $\mathbf{v}_p = [v_p^{00}(1,1), v_p^{01}(1,1), v_p^{02}(1,1), v_p^{03}(1,1), v_p^{10}(1,1), \cdots, v_p^{33}(1,1)]^T$ the transform matrix $\mathbf{M}$ is

$$\mathbf{M} = \begin{bmatrix} \mathbf{M}_0 & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{M}_0 & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & -\mathbf{M}_0 & 2\mathbf{M}_0 & \mathbf{0} \\ \mathbf{0} & \mathbf{M}_0 & -6\mathbf{M}_0 & 6\mathbf{M}_0 \end{bmatrix} \quad \text{where} \quad \mathbf{M}_0 = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & -1 & 2 & 0 \\ 0 & 1 & -6 & 6 \end{bmatrix}$$

Using the above method, we give an algorithm for the fast computation of 3-D moments:

INPUT A list of background-to-object transitions is obtained by a surface tracking algorithm [2]. For each of the transitions, a position coordinate (x, y, z) and the direction component Δx are recorded.

OUTPUT A list of geometric moments m_{pqr} for $0 \leq p + q + r \leq K$.

DATA STRUCTURE $(K + 1)$ integer arrays of size $N \times N$ and one integer array of size N, all initialized to zero. They are used to store $v_p^{qr}(y, z)$. The reuse of the storage has been considered.

ALGORITHM

1. for all transitions when $\Delta x \neq 0$ do
   for $p = 0, \cdots, K$ do
      if $\Delta x = -1$ then $v_p(y, z) = v_p(y, z) - x^{p+1}$
      if $\Delta x = 1$ then $v_p(y, z) = v_p(y, z) + x^{p+1}$
2. for $p = 0, \cdots, K$ do
   for $q = 0, \cdots, K - p$ do
      compute $v_p^q(y, z)$ from $v_p(y, z)$ by the Hatamian filtering
      for $r = 0, \cdots, K - p - q$ do
         compute $v_p^{qr}(1, z)$ from $v_p(1, z)$ by the Hatamian filtering
3. compute the moment m_{pqr} as a linear combination of $v_p^{ij}(1,1)$ for $i = 0, \cdots, q$ and $j = 0, \cdots, r$ given by Eqs. (10) and (17)

For a convex object in an $N \times N \times N$ image, there are at most $2N^2$ transitions of which $\Delta x \neq 0$. Assume that the number of such transitions is $2N^2$. Then in step 1 we need $2KN^2$ multiplications and $2(K + 1)N^2$ additions. In step 2 we need $(\frac{1}{2}K^2 + \frac{3}{2}K + 1)N^2$ additions for all the y-directional filterings and $(\frac{1}{6}K^3 + K^2 + \frac{11}{6}K + 1)N$ additions for all the z-directional filterings. So, our algorithm is of $O(N^2)$, requiring $O(2KN^2)$ multiplications and $O(\frac{1}{2}K^2N^2)$ additions. Compared to the earlier methods [1, 4, 6], our method improves the computational efficiency significantly (see Table 1).

4 Discussion and Conclusion

The geometric moments of 2-D and 3-D objects have been widely used in many image analysis and pattern recognition tasks. However, the computation of the moments requires a large amount of computing operations, especially when a straightforward method is used. In the 2-D case, Li and Shen [5] proposed to

Table 1. A comparison of complexity in computing all the moments of order up to K from a discrete 3-D image of size $N \times N \times N$.

Method	Multiplication	Addition
Straightforward	$\left(\frac{1}{6}K^3 + K^2 + \frac{11}{6}K + 1\right) N^3$	$\left(\frac{1}{6}K^3 + K^2 + \frac{11}{6}K + 1\right) N^3$
Cyganski *et al.* [1]	$\left(\frac{1}{2}K^2 - \frac{1}{2}K\right) N^2$	$(K+1)N^3 + \left(\frac{1}{2}K^2 + \frac{7}{2}K + 4\right) N^2$
Li and Shen [6]	0	$\left(\frac{1}{2}K^4 + \frac{1}{2}K^3 + 1\right) N^3$
Li and Ma [4]	$2KN^2$	$N^3 + (K^2 + 2K)N^2$
Ours	$2KN^2$	$\left(\frac{1}{2}K^2 + \frac{7}{2}K + 3\right) N^2$

use Green's theorem for fast compuation. However, this method is not accurate since the continuous version of Green's theorem is applied to a discrete image directly. Yang and Albregtsen [9, 10] proposed to use a discrete Green's theorem so that the exact value of a double sum is obtained. Fast computation of 3-D moments have been proposed in a number of previous papers [1, 4, 6]. All these methods require computation of $O(N^3)$. Using the discrete Gauss' theorem, we have reduced the complexity to $O(N^2)$. The discrete Gauss' theorem improves the computational efficiency and at the same time gives the exact value of a triple sum.

References

1. Cyganski, D., Kreda, S. J., Orr, J. A.: Solving for the general linear transformation relating 3-D objects from the minimum moments. Proc. SPIE Vol. 1002, Intelligent Robots and Computer Vision VII (1988) 204–211
2. Gordon, D., Udupa, J. K.: Fast surface tracking in three-dimensional binary images. Comput. Vision Graph. Image Process. **45** (1989) 196–214
3. Hatamian, M.: A real-time two-dimensional moment generating algorithm and its single chip implementation. IEEE Trans. ASSP **34** (1986) 546–553
4. Li, B.-C., Ma, S. D.: Efficient computation of 3D moments. Proc. 12th Int. Conf. Pattern Recogn., Vol. I (1994) 22–26
5. Li, B.-C., Shen, J.: Fast computation of moment invariants. Pattern Recogn. **24** (1991) 807–813
6. Li, B.-C., Shen, J.: Pascal triangle transform approach to the calculation of 3D moments. CVGIP: Graphical Models and Image Processing **54** (1992) 301–307
7. Lo, C.-H., Don, H.-S.: 3-D moment forms: their construction and application to object identification and positioning. IEEE Trans. Pattern Anal. Machine Intell. **11** (1989) 1053–1064
8. Pei, S.-C., Liou, L.-G.: Using moments to acquire the motion parameters of a deformable object without correspondences. Image and Vision Computing **12** (1994) 475–485
9. Yang, L., Albregtsen, F.: Fast computation of invariant geometric moments: a new method giving correct results. Proc. 12th Int. Conf. Pattern Recognition, Vol. I (1994) 201–204
10. Yang, L., Albregtsen, F.: Fast and exact computation of Cartesian geometric moments using discrete Green's theorem. Submitted to Pattern Recogn. (1994)