

Refined Method for the Fast and Exact Computation of Moment Invariants

Humberto Sossa¹ and Jan Flusser²

¹ Centro de Investigación en Computación - IPN
Av. Juan de Dios Bátiz s/n, Esquina con Miguel Othón de Mendizábal
Colonia Nueva Industrial Vallejo, C. P. 07700, México, D. F. Mexico

² Institute of Information Theory and Automation
Academy of Sciences of the Czech Republic
Pod vodárenskou veží 4. 182 08 Prague 8, Czech Republic

Abstract. Geometric moments have been proven to be a very efficient tool for description and recognition of binary shapes. Numerous methods for effective calculation of image moments have been presented up to now. Recently, Sossa, Yañez and Díaz [Pattern Recognition, 34(2):271-276, 2001] proposed a new algorithm based on a morphologic decomposition of the image into a set of closed disks. Their algorithm yields approximative results. In this paper we propose a refinement of their method that performs as fast as the original one but gives exact results.

1 Introduction

Image moments and various types of moment-based invariants play a very important role in object recognition and shape analysis [3], [1]. The $(p+q)$ th order Cartesian geometric moment M_{pq} of a two-dimensional grey-level image $f(x, y)$ (for short 2-D moment) is defined as

$$M_{pq} = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} x^p y^q f(x, y) dx dy \quad p, q = 0, 1, 2, \dots \quad (1)$$

In a binary image the characteristic function takes only the values 1 or 0, supposing that for the interest region R holds $f(x, y) = 1$. Thus, the definition simplifies to the form

$$M_{pq} = \iint_R x^p y^q dx dy, \quad (2)$$

In the discrete case, the double integral must be replaced by a summation. The most common way how to do that is to employ the rectangular (i.e. zero-order) method of numeric integration. Then (1) turns to the well-known form

$$m_{pq} = \sum_{x=1}^N \sum_{y=1}^N x^p y^q f_{ij}, \quad (3)$$

where N is the size of the image and f_{ij} are the grey levels of individual pixels $(x, y) \in Z^2$. Finally, for binary region R we get

$$m_{pq} = \sum_{(x,y) \in R} x^p y^q. \quad (4)$$

Fast and exact computation of 2-D geometric moments of binary objects has been considered as an important problem for its importance in numerous image-processing applications.

Since direct calculation of discrete moments from eq. (4) is time-consuming (it requires $O(pqN^2)$ operations), a large amount of effort has been spent in the last decade to develop more effective algorithms.

Recently, Flusser [2] pointed out that the formula (4), commonly used in the literature for the calculation of the discrete moments of a rectangular block yields inaccurate results. In the same paper, Flusser presented a correction of the method originally proposed by Spiliotis and Mertzios [6].

In this paper we show that the recent method by Sossa et al. [5] also suffers by this inaccuracy and we propose how to correct it. We use the same block-wise image representation as Sossa et al. proposed in [5] but we present a different scheme to calculate the block moments. We show that the new method yields more accurate results than the original one.

2 Original Method by Sossa et al.

In [5], a novel approach to the calculation of geometric moments of binary image was introduced. This method was based on morphologic erosion of the original shape.

The method performs in three basic steps:

1. Decompose the given image into a union of disjoint disks;
2. Compute the geometric moments for each of these disks;
3. Obtain the final moments as a sum of the moments computed for each disk.

Various metrics defined on a discrete plane can be employed in Step 1. The use of different metrics leads to different image decompositions. Sossa et al. used d_8 "maximum" metric defined as

$$d_8(P_1, P_2) = \max\{|x_1 - x_2|, |y_1 - y_2|\} \quad (5)$$

where P_i is a point whose coordinates are (x_i, y_i) . It should be noted that a circular disk in d_8 metric is a square in Euclidean metric. The basic algorithm used by Sossa et al. to obtain the desired decomposition of a binary region $R \subset Z^2$ iteratively erodes R until the maximal disk completely contained in the original region R is obtained. Then this disk is eliminated from the region R and the remaining shape is assigned to R . The algorithm repeats this procedure until R becomes empty.

To calculate the moments of a disk D_p^r of radius r with center (X_c, Y_c) in $p \in Z^2$, Sossa et al. used eq. (4). The explicit formulae for the first ten geometric moments of this disk in d_8 metric are:

$$m_{00} = (2r + 1)^2 \quad m_{10} = m_{00}X_c \quad m_{01} = m_{00}Y_c \quad m_{11} = m_{10}Y$$

$$\begin{aligned}
 m_{20} &= \frac{m_{00}}{3} (3X_c^2 + r(r+1)) & m_{02} &= \frac{m_{00}}{3} (3Y_c^2 + r(r+1)) \\
 m_{21} &= m_{20} Y_c & m_{12} &= m_{02} X_c \\
 m_{30} &= m_{10} (X_c^2 + r(r+1)) & m_{03} &= m_{01} (Y_c^2 + r(r+1))
 \end{aligned} \tag{6}$$

To obtain these expressions, Sossa et al. used the well-known formulae for sums of integer powers:

$$\sum_{k=1}^n k = \frac{n}{2}(n+1), \quad \sum_{k=1}^n k^2 = \frac{n}{6}(n+1)(2n+1), \quad \sum_{k=1}^n k^3 = \left[\frac{n}{2}(n+1) \right]^2.$$

For details on the Sossa et al. method we refer to [5].

3 Refined Method

In this Section, we present a refinement of Sossa et al. method, which uses different formulae for calculation of disk moments. To explain the idea, we recall the original definition of moments in the continuous domain:

$$M_{pq} = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} x^p y^q f(x, y) dx dy \tag{7}$$

where $f(x, y)$ is, in this case, the characteristic function of a disk. Clearly, eq. (4) is only an approximation of eq. (7). An error $|M_{pq} - m_{pq}|$ is introduced due to the zero-order approximation and numeric integration of $x^p y^q$ over each pixel of the disk. Below we show how to calculate the geometric moments of a disk exactly without any approximation.

Given a closed disk D_p^r of radius r with center (X_c, Y_c) in $p \in Z^2$ with corner pixels centered in $(X_c - r, Y_c - r)$, $(X_c + r, Y_c - r)$, $(X_c - r, Y_c + r)$ and $(X_c + r, Y_c + r)$. The characteristic function of this disk is

$$f(x, y) = \begin{cases} 1 & \text{if } (x, y) \in (X_c - r - 0.5, X_c + r + 0.5) \times (Y_c - r - 0.5, Y_c + r + 0.5) \\ 0 & \text{otherwise.} \end{cases}$$

According to eq. (7), the exact moments of the disk D_p^r are given as

$$\begin{aligned}
 M_{pq} &= \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} x^p y^q f(x, y) dx dy \\
 &= \frac{1}{(p+1)(q+1)} \left[(X_c + r + 0.5)^{p+1} - (X_c - r - 0.5)^{p+1} \right] \cdot \\
 &\quad \left[(Y_c + r + 0.5)^{q+1} - (Y_c - r - 0.5)^{q+1} \right]
 \end{aligned} \tag{8}$$

The reader can easily see that the corrected set of expressions for the first ten moments of the closed disk D_p^r becomes:

$$\begin{aligned}
 M_{00} &= (2r+1)^2 & M_{10} &= M_{00}X_c & M_{01} &= M_{00}Y_c & M_{11} &= M_{10}Y \\
 M_{20} &= \frac{M_{00}}{3}(3X_c^2 + r(r+1) + 0.25) & M_{02} &= \frac{M_{00}}{3}(3Y_c^2 + r(r+1) + 0.25) \\
 M_{21} &= M_{20}Y_c & M_{12} &= M_{02}X_c \\
 M_{30} &= M_{10}(X_c^2 + r(r+1) + 0.25) & M_{03} &= M_{01}(Y_c^2 + r(r+1) + 0.25)
 \end{aligned} \tag{9}$$

4 Comparison of the Two Methods

While eq. (8) provides exact results, eq. (4) calculates some moments with errors due to the approximation. It can be shown that there is always $M_{pq} \geq m_{pq}$; the error $M_{pq} - m_{pq}$ depends on the values of p and q . Comparing eqs. (9) and (6), we can evaluate this errors easily, for instance

$$\begin{aligned}
 M_{20} - m_{20} &= \frac{1}{3}(6X_c^2r + 3X_c^2 + 2r^3 + 3r^2 + 1.5r + 0.25)(2r+1) - \\
 &\quad \frac{1}{3}(6X_c^2r + 3X_c^2 + 2r^3 + 3r^2 + r)(2r+1) \\
 &= \frac{1}{3}(2r+1)(0.5r + 0.25) \\
 &= \frac{1}{12}m_{00}.
 \end{aligned}$$

Similarly,

$$\begin{aligned}
 M_{02} - m_{02} &= \frac{m_{00}}{12}, \quad M_{30} - m_{30} = \frac{m_{10}}{4}, \quad M_{21} - m_{21} = \frac{m_{01}}{12}, \\
 M_{12} - m_{12} &= \frac{m_{10}}{12}, \quad M_{03} - m_{03} = \frac{m_{01}}{4}.
 \end{aligned} \tag{10}$$

On the other hand, for some moments both methods produce exact results:

$$M_{00} - m_{00} = 0, \quad M_{01} - m_{01} = 0, \quad M_{10} - m_{10} = 0, \quad M_{11} - m_{11} = 0. \tag{11}$$

5 Impact on the Values of the Invariants

In object recognition we rarely use directly the geometric moments as the features because they are sensitive to particular position of the object in the image plane. Instead, we employ certain functions of moments, called moment invariants, that are

invariant to expected intraclass variations of the objects. Typically, shift and rotation invariance is required in many practical tasks. In this Section, we analyze how the values of various moment invariants differ depending on if m_{pq} or M_{pq} are used.

Probably the most popular set of moment invariants was derived by Hu [3]. These features are invariant to translation and rotation of the object. In continuous domain the first two are defined as

$$\phi_1 = \mu_{20} + \mu_{02}, \phi_2 = (\mu_{20} - \mu_{02})^2 - 4\mu_{11} \quad (12)$$

where

$$\mu_{pq} = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} (x - x_t)^p (y - y_t)^q f(x, y) dx dy \quad (13)$$

is the central moment of the object $f(x, y)$ and (x_t, y_t) are the coordinates of the object centroid. Hu also showed that scaling invariance can be achieved via normalization of each central moment by $\mu_{00}^{(p+q+2)/2}$.

It can be easily observed that both algorithms give the same position of object centroid:

$$x_t \equiv \frac{m_{10}}{m_{00}} \equiv \frac{M_{10}}{M_{00}} \text{ and } y_t \equiv \frac{m_{01}}{m_{00}} \equiv \frac{M_{01}}{M_{00}},$$

Thus, central moments are affected exactly in the same way as geometric moments. However, since μ_{10} and μ_{01} equal zero by definition, the relations (10) simplify to the form

$$M_{20} - \mu_{20} = \frac{m_{00}}{12}, M_{02} - \mu_{02} = \frac{m_{00}}{12},$$

$$M_{30} - \mu_{30} = 0, M_{21} - \mu_{21} = 0, M_{12} - \mu_{12} = 0, M_{03} - \mu_{03} = 0$$

where μ_{pq} denotes the central moments calculated by traditional method while M_{pq} stands for the central moments calculated by the refined method.

Now we can observe an interesting fact: The only Hu's invariant whose values depend on the method of calculation is the first one. Clearly,

$$\phi_1(M_{pq}) - \phi_1(\mu_{pq}) = \frac{\mu_{00}}{6}.$$

ϕ_3 , ϕ_4 , ϕ_5 and ϕ_7 contains only 3rd-order moments, so they cannot be affected. ϕ_2 and ϕ_6 stay also constant due to the error cancellation effect. The same observation can be found in [4].

Another set of invariants called affine moment invariants (AMI's) was proposed by Flusser and Suk [1]. They are more general than Hu's invariants because they are invariant not only to rotation and scaling but to general affine transformation. Thanks to this, they can be successfully used in object recognition on images deformed by slant or anisotropic scaling. The first six AMI's are shown below.

$$\begin{aligned}
I_1 &= (\mu_{20}\mu_{02} - \mu_{11}) / \mu_{00}^4 \\
I_2 &= (\mu_{30}^2\mu_{03}^2 - 6\mu_{30}\mu_{21}\mu_{12}\mu_{03} + 4\mu_{30}\mu_{12}^2 + 4\mu_{21}^2\mu_{03} - 3\mu_{21}^2\mu_{12}^2) / \mu_{00}^{10} \\
I_3 &= (\mu_{20}(\mu_{21}\mu_{03} - \mu_{12}^2) - \mu_{11}(\mu_{30}\mu_{03} - \mu_{21}\mu_{12}) + \mu_{02}(\mu_{30}\mu_{12} - \mu_{21}^2)) / \mu_{00}^7 \\
I_4 &= (\mu_{20}^3\mu_{03}^2 - 6\mu_{30}\mu_{11}\mu_{12}\mu_{03} - 6\mu_{20}^2\mu_{02}\mu_{21}\mu_{03} + 9\mu_{20}^2\mu_{02}\mu_{12}^2 \\
&\quad + 12\mu_{20}\mu_{11}^2\mu_{21}\mu_{03} + 6\mu_{20}\mu_{11}\mu_{02}\mu_{30}\mu_{03} - 18\mu_{20}\mu_{11}\mu_{02}\mu_{21}\mu_{12} \\
&\quad - 8\mu_{11}^3\mu_{30}\mu_{03} - 6\mu_{20}\mu_{02}^2\mu_{30}\mu_{12} + 9\mu_{20}\mu_{02}^2\mu_{21}^2 + 12\mu_{11}^2\mu_{02}\mu_{30}\mu_{12} \\
&\quad - 6\mu_{11}\mu_{02}^2\mu_{30}\mu_{21} + \mu_{02}^3\mu_{30}^2) / \mu_{00}^{11} \\
I_5 &= (\mu_{40}\mu_{04} - 4\mu_{31}\mu_{13} + 3\mu_{22}^2) / \mu_{00}^6 \\
I_6 &= (\mu_{40}\mu_{04}\mu_{22} + 2\mu_{31}\mu_{22}\mu_{13} - \mu_{40}\mu_{13}^2 - \mu_{04}\mu_{31}^2 - 3\mu_{22}^2) / \mu_{00}^9
\end{aligned}$$

We can easily prove that, for the same reason as in the previous case, I_2 stays the same irrespective of the calculation algorithm. However, other AMI's change if M_{pq} are used instead of μ_{pq} . We can observe that, for instance,

$$I_1(M_{pq}) - I_1(\mu_{pq}) = \frac{\mu_{00}}{12}(\mu_{20} - \mu_{02}) + \frac{\mu_{00}^2}{144}.$$

Similar (but more complicated) relations can be derived for other AMI's too.

Summarizing the above analysis, we can conclude that when using Hu's invariants, one does not need to use the refined algorithm for moment calculation (with exception of ϕ_1). On the other hand, the AMI's should be computed using the refined method if accurate values are required.

6 Numerical Experiments

In this Section, we experimentally demonstrate the difference between classical and new formula for moment calculation. From the theoretical analysis given above implies that some moments and moment invariants are identical regardless of the fact whether eq. (4) or eq. (8) is used. In this experiment we show the differences between second-order central moments, first Hu's invariant ϕ_1 and six affine moment invariants I_1, \dots, I_6 for three binary images.

In Fig. 1, one can see our test objects called "Plane", "Duck", and "Snake", respectively. In Table 1, the relative errors of the moments and of the invariants are shown. The errors we have investigated are defined as follows:

$$\begin{aligned}
e_1 &= \frac{M_{20} - \mu_{20}}{M_{20}}, \quad e_2 = \frac{M_{02} - \mu_{02}}{M_{02}}, \quad e_3 = \frac{\phi_1(M_{02}) - \phi_1(\mu_{02})}{\phi_1(M_{02})}, \\
e_{j+3} &= \frac{I_j(M_{02}) - I_j(\mu_{02})}{I_j(M_{02})}, \quad j = 1, \dots, 6.
\end{aligned}$$



Fig. 1. Test objects used in the experiment: Plane(left), Duck(middle), and Snake(right).

Table 1. Relative differences between traditional and new formulae for moment calculation. For simplicity, all values were multiplied by 10^4 .

	e_1	e_2	e_3	e_4	e_5	e_6	e_7	e_8	e_9
Plane	0.16	0.23	0.19	0.40	0	0.21	0.63	0.91	1.63
Duck	0.19	0.41	0.26	0.81	0	0.41	1.24	2.08	3.27
Snake	0.18	54.89	0.37	56.73	0	-6.52	2.02	2.21	145.07

The most important observation that is based on this experiment can be summarized in the following way. For elongated objects (where elongation is parallel to one of the axes) the relative errors can be high and might, in some cases, cause misclassification. An extreme case is a one-pixel thick horizontal line, for which $\mu_{02} = 0$ but M_{02} is proportional to its length. Relative errors of other objects are much smaller, as can be seen in Table 1 when comparing Plane or Duck to Snake. Thus, when dealing with elongated objects, it is recommended to use the refined formula (8) for moment calculation. On the other hand, for objects having "regular" shapes traditional computation according to eq. (4) can be used without significant loss of accuracy.

7 Conclusion

In this paper, we presented a new method for calculating geometric moments and, consequently, of moment invariants of binary objects. This method is a refinement of Sossa et al. algorithm that was based on a decomposition of the object into disjoint disks. In the original paper by Sossa et al., zero-order approximation was used for numerical integration when calculating moments of the disks. In this paper we proposed an exact formula with no approximation. We analyzed (both theoretically and experimentally) the differences between these two formulae and the influence of the refined method on the values of moment invariants. We demonstrated that for some shapes the new method yields a significant increase in accuracy.

Acknowledgment

Financial support of this research was provided by the National Council of Science and Technology of Mexico (CONACYT) under projects No. 41529 and 40114Y, by the National Polytechnic Institute of Mexico (IPN) under project No. 20030658 and by the Grant Agency of the Czech Republic under the project No. 102/00/1711.

References

1. J. Flusser and T. Suk, Pattern recognition by affine moment invariants, *Pattern Recognition*, 26(1):167-174, 1993.
2. J. Flusser, Refined moment calculation using image block representation, *IEEE Transactions on Image Processing*, 9(11):1977-1978, 2000.
3. M. K. Hu, Visual pattern recognition by moment invariants, *IRE Transactions on Information Theory*, 8:179-187, 1962.
4. W. G. Lin and S. S. Wang, A note on the calculation of moments, *Pattern Recognition Letters*, 15(11):1065-1070, 1994.
5. H. Sossa, C. Yañez and J. L. Díaz, Computing geometric moments using morphological erosions, *Pattern Recognition*, 34(2):271-276, 2001.
6. I. M. Spiliotis and B. G. Mertzios, Real-time computation of 2-D moments on binary images using image block representation, *IEEE Transactions of Image Processing*, 7(11):1609-1615, 1998.