

Parameterized real-time moment computation on gray images using block techniques

Iraklis M. Spiliotis · Yiannis S. Boutalis

Received: 30 April 2009 / Accepted: 18 October 2009 / Published online: 17 November 2009
© Springer-Verlag 2009

Abstract Moments constitute a well-known tool in the field of image analysis and pattern recognition, but they suffer from the drawback of high computational cost. Efforts for the reduction of the required computational complexity have been reported, mainly focused on binary images, but recently some approaches for gray images have been also presented. In this study, we propose a simple but effective approach for the computation of gray image moments. The gray image is decomposed in a set of binary images. Some of these binary images are substituted by an ideal image, which is called “half-intensity” image. The remaining binary images are represented using the image block representation concept and their moments are computed fast using block techniques. The proposed method computes approximated moment values with an error of 2–3% from the exact values and operates in real time (i.e., video rate). The procedure is parameterized by the number m of “half-intensity” images used, which controls the approximation error and the speed gain of the method. The computational complexity is $O(kL^2)$, where k is the number of blocks and L is the moment order.

Keywords Image analysis · Gray image · Moments · Image block representation · Real-time computation

1 Introduction

Moments and functions of moments have been successfully and extensively used in pattern recognition, image classification and scene analysis applications [1–9]. Moreover, image moments have also been used for facial expression recognition [10], stereo image matching [11], image retrieval from databases [12] and image watermarking [13].

The geometric moment of order (p, q) of a 2D function $g(u, v)$ is defined as:

$$m_{pq} = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} g(u, v) u^p v^q \, dudv \quad (1)$$

Given an $N_1 \times N_2$ discrete image $g(x, y)$, the moment of order (p, q) is defined as:

$$m_{pq} = \sum_{x=0}^{N_1-1} \sum_{y=0}^{N_2-1} x^p y^q g(x, y), \quad p, q \in \mathbb{N}_0. \quad (2)$$

The lower-order moments represent several well-known geometric properties of the distribution, including area, center of mass, principal axes, radius of gyration, skewness and kurtosis [3]. Hu’s moment invariants [1], comprise a seven-moment set, which is based on low order ($p + q \leq 3$) normalized central moments and are invariant to translation, rotation and scale change. In pattern recognition applications, a small set of lower-order moments are able to discriminate among different patterns [14], and low-order geometric moments and their derivatives (i.e., central, normalized central and moment invariant) have been extensively used in these applications. High-order moments have been used for the inverse problem, of reconstructing the original image from a finite set of moment values [15], where orthogonal-based moments instead of geometric ones are usually used.

I. M. Spiliotis (✉) · Y. S. Boutalis
Department of Electrical and Computer Engineering,
Democritus University of Thrace, 67100 Xanthi, Greece
e-mail: spiliot@ee.duth.gr

Y. S. Boutalis
e-mail: ybout@ee.duth.gr

The computation of moments up to the order $(L - 1, L - 1)$, directly from (2) is of order N^2L^2 and specifically requires $2N^2L^2$ power calculations, $2N^2L^2$ multiplications and $(N^2 - 1)L^2$ additions. Several researchers have proposed approaches for the reduction of the required computational complexity, but the majority of these methods are mainly focused on binary images. Li and Shen [16] proposed the use of Green's theorem in continuous domain, which evaluates the double integral over the object by means of single integration along the object boundary. Jiang and Bunke [17] proposed a polygonal approximation of the object boundary and then the computation of moments using Green's theorem. Yang and Albreghsen [14] used a discrete version of Green's theorem for the computation of image moments. Zakaria et al. [18], Dai et al. [19], Li [20] and Flusser [21] proposed various approaches based on the decomposition of the object into rows or row segments.

Spiliotis and Mertzios [22] proposed a binary image representation method that represents the image in a form suitable for the fast implementation of various binary image-processing algorithms. This binary image representation scheme is called image block representation (IBR), since it represents the image as a set of non-overlapping homogeneous rectangular areas. The most important characteristic of the IBR is that a perception of image areas greater than a pixel is provided to the machine and, therefore, all the operations on the pixels belonging to a block may be substituted by a simple operation on the block. Based on this representation, the real-time moment computation on binary images using the coordinates of the upper left and lower right corner of each block is proposed in [22]. Other algorithms implemented on block represented binary images are: fast shift, scale and rotation of binary images [23], connected component labeling [23], logic operations [23], a fast parallel skeletonization algorithm [24], a fast thinning algorithm [25] and a fast algorithm for the computation of the Hough transform [26].

Chung and Chen [27] extended the idea in [22] to gray images by separating gray images to "homogeneous" blocks of image intensities and performing moment calculation using the blocks. The definition of homogeneity introduces an error in the moment calculation requiring a compromise between the accuracy and the number of homogeneous blocks, making the error smaller when a large number of homogeneous blocks are used.

Recently, Hosny [28] proposed a method for the calculation of gray image moments, which provides exact moment values and operates fast. Hosny's method uses a correction proposed by Liao and Pawlak [15], for the transition from the continuous to the discrete space without introducing any errors. In the sequel, exploiting the

rectangular form of the pixel that is introduced by the correction, Hosny's method separates the 2D computation in two steps, by successive computation of 1D moments for each image row.

In this study, the gray-level image is decomposed in a set of binary images (bitplanes). It is proved experimentally that the lower-order bitplanes are sufficiently similar to an ideal image, called "half-intensity" image. Therefore, the moments of the lower-order bitplanes are substituted with the moments of the "half-intensity" image. Moreover, the computation of the moments of the higher-order bitplanes, using the block-based method of [22], results in the real-time moment computation. This method permits the computation of the moments with only a few higher-order binary images and keeps the approximation error low, while the proposed algorithm operates in real time. A preliminary version of this work has been presented in [29].

The remainder of the paper is organized as follows. Section 2 gives the basic definitions for the block representation of binary images and presents the mathematical formulae for the real-time computation of moments on binary images. Section 3 presents the decomposition of a gray image to a set of binary images. In Sect. 4, the computation of the moments is given, while Sect. 5 provides experimental results, an analysis of the computational complexity and comparisons. Finally, Sect. 6 provides some concluding remarks.

2 IBR and moments computation on binary images

Suppose that in a binary image the object pixels are assigned to level 1. Suppose also that the object pixels are represented by a set of non-overlapping rectangles with edges parallel to the axes containing integer number of pixels, in such a way that every object pixel belongs to only one block. These rectangles are called blocks. It is always feasible to represent a binary image with a set of all the nonoverlapping blocks with object level and this representation is called IBR [23].

The IBR concept leads to a simple and fast algorithm, which requires just one pass of the image and simple bookkeeping process. As a result of the application of the above algorithm, we obtain a set of all the rectangular areas with level 1 that form the object. A block-represented image is denoted as $f(x, y) = \{b_i : i = 0, 1, \dots, k - 1\}$, where k is the number of the blocks. Each block is described by four integers, the coordinates of the upper left and lower right corner in vertical and horizontal axes, $b_i = (x_{1,b_i}, x_{2,b_i}, y_{1,b_i}, y_{2,b_i})$. The block extraction process is implemented easily with low computational complexity, since it is a pixel checking process without numerical

operations. The block extraction process requires a pass from each line y of the image. In this pass, all object-level intervals are extracted and compared with the previously extracted blocks. In the following, an IBR algorithm is given.

Algorithm 1: Image block representation.

Step 1: Consider each line y of the image f and find the object-level intervals in line y .

Step 2: Compare intervals and blocks that have pixels in line $y - 1$.

Step 3: If an interval does not match with any block, this is the beginning of a new block.

Step 4: If a block matches with an interval, the end of the block is in the line y .

Given a specific binary image, different sets of different blocks can be formed. Actually, the nonunique block representation does not have any implications on the implementation of any operation on a block-represented image. The optimum representation is characterized by the minimum possible number of blocks.

Since the background level is 0, only the pixels with level 1 are taken into account for the computation of the moments. Thus, the 2D geometrical moments of order (p, q) of the image $f(x, y)$ are defined by the relation:

$$m_{pq} = \sum_x \sum_y x^p y^q \quad \forall x, y : f(x, y) = 1. \tag{3}$$

Since the image is represented by blocks, all the object pixels belong to the k blocks and, taking into account the rectangular form of the block, (3) may be rewritten [19] as:

$$\begin{aligned} m_{pq} &= \sum_{i=0}^{k-1} m_{pq}^{b_i} = \sum_{i=0}^{k-1} \sum_{x=x_{1b_i}}^{x_{2b_i}} \sum_{y=y_{1b_i}}^{y_{2b_i}} x^p y^q \\ &= \sum_{i=0}^{k-1} \left(\sum_{x=x_{1b_i}}^{x_{2b_i}} x^p \right) \left(\sum_{y=y_{1b_i}}^{y_{2b_i}} y^q \right) = \sum_{i=0}^{k-1} S_{x_{1b_i}, x_{2b_i}}^p S_{y_{1b_i}, y_{2b_i}}^q \end{aligned} \tag{4}$$

where

$$\begin{aligned} S_{1,n}^1 &= \frac{n(n+1)}{2}, \quad S_{1,n}^2 = \frac{n(n+1)(2n+1)}{6}, \quad S_{1,n}^3 = \frac{n^2(n+1)^2}{4}, \\ S_{1,n}^4 &= \frac{n(n+1)(2n+1)(3n^2+3n+1)}{30} \\ \sum_{j=1}^m \binom{m+1}{j} S_{1,n}^j &= (n+1)^{m+1} - (n+1). \end{aligned} \tag{5}$$

Note that using (4), the computational complexity is reduced to $O(N)$ instead of $O(N^2)$ using (2). Moreover, the summations of x^p and y^q may be computed using the above known formulae (5) resulting in further speed-up of the calculations.

As shown in [22], this method has a computational complexity of $O(k)$, where k is the number of blocks and the method operates in real time. It has been shown also that other sets of statistical moments as central moments, normalized central moments and Hu’s moment invariants are computed in real time since are based on geometric moments.

As shown by Liao and Pawlak [15], (2) is not a very exact approximation of (1), due to the zero-order approximation of the double integral of (1) with the summations of (2) and due to the numeric integration of $x^p y^q$ over each pixel and proposed a correction for the moments computation at the discrete domain.

In accordance, Flusser [30] proposed a refinement to [22], which concludes that the moment of order (p, q) of the block b is given by:

$$m_{pq}^b = \frac{(x_2 + 0.5)^{p+1} - (x_1 - 0.5)^{p+1}}{p + 1} \cdot \frac{(y_2 + 0.5)^{q+1} - (y_1 - 0.5)^{q+1}}{q + 1} \tag{6}$$

where it was also proposed that parts of (6) could be pre-calculated, something which may also be done to quantities of (5).

3 Decomposition of the gray image

Consider the gray image $g(x, y)$, with 2^n gray levels and dimensions $N_1 \times N_2$. The gray image can be decomposed into a set of n binary images, so that each binary image is a bitplane of the original gray image. To do that, each pixel value of the gray image is binary represented by n bits. Each one of these n bits is the corresponding pixel value of the resulting n binary images. If for example each gray image pixel requires 1 byte, then the gray image is decomposed in eight binary images. This way, the first binary image b_{n-1} contains the most significant bits (MSB) of the binary representation of the image intensities $g(x, y)$, the second binary image b_{n-2} contains the second MSB and so on until the binary image b_0 , containing the least significant bits of the binary representation of the gray image intensities. The relation between the gray image $g(x, y)$ and the n binary images b_{n-1}, \dots, b_1, b_0 is given by:

$$\begin{aligned} g(x, y) &= 2^{n-1} b_{n-1}(x, y) + \dots + 2^0 b_0(x, y) \\ &= \sum_{i=0}^{n-1} 2^i b_i(x, y). \end{aligned} \tag{7}$$

Figure 1 illustrates a set of test images with 256 gray levels. Figure 2 illustrates the decomposition of the image of Fig. 1h to the eight corresponding binary images. It can be observed that lower-order binary images look quite

Fig. 1 A set of test images with 256 gray levels. **a–f** 256 × 256 pixels. **g–i** 200 × 200 pixels



noisy. This feature is exploited below for the reduction of the computational cost of moment calculations.

4 Computation of moments on gray images

In this section, the real-time computation of moments on gray images is presented.

4.1 Geometric moments

Applying (7) to (2) for the computation of moments, the following formula is obtained:

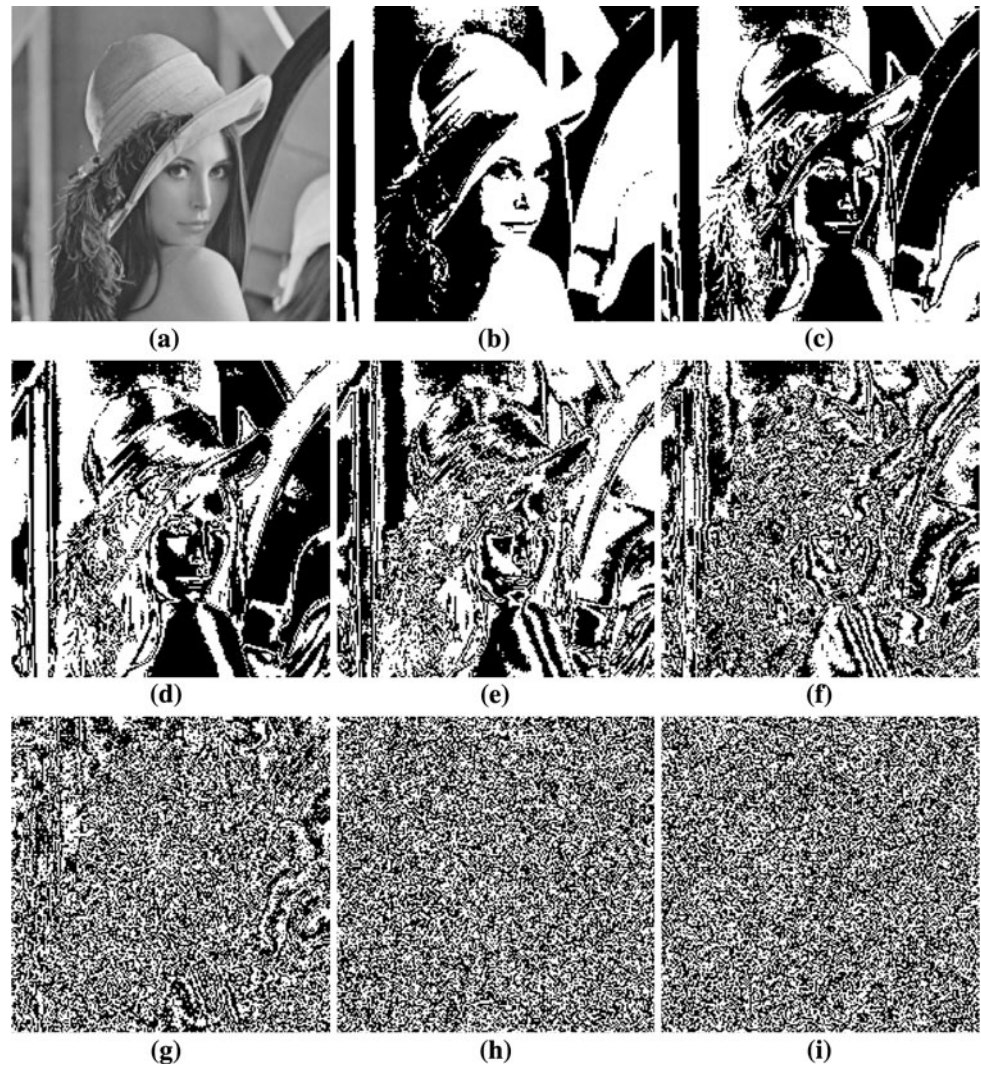
$$\begin{aligned}
 m_{pq} &= \sum_{x=0}^{N_1-1} \sum_{y=0}^{N_2-1} x^p y^q g(x, y) \\
 &= \sum_{x=0}^{N_1-1} \sum_{y=0}^{N_2-1} x^p y^q \left[\sum_{i=0}^{n-1} 2^i b_i(x, y) \right] = \sum_{i=0}^{n-1} 2^i m_{ipq} \quad (8)
 \end{aligned}$$

where $mb_{i,pq}$ is the (p, q) th order moment of the binary image b_i that derives from the decomposition of the gray image. The binary images are represented by blocks and their moments are fast computed using (4) and Flusser’s exact formula [30] of (6) with pre-calculations. Therefore, (8), provides a fast method for the calculation of the exact moment values.

At this point, two observations can be made. The first observation is that as (8) implies, the moments of lower-order bitplanes do not participate equally in the computation of the gray image moments, because of the weight factors of 2^i . If a number of these bitplanes are omitted from the calculation of (8), this results in calculation error and computational savings.

As illustrated in Fig. 2, the lower-order bitplanes look like noise with transitions from white to black. The second observation is that the lower-order bitplanes are similar to an image with intensity equal to the half of the object level. This “half-intensity” image may be a chessboard (or

Fig. 2 **a** The original gray image with 256 gray levels. **b–i** The decomposition of the original image results in these eight binary images b_7 to b_0 , where b_7 at **b** derives from the most significant bits and b_0 at **i** derives from the least significant bits of the pixel values of **a**



checkerboard) image, or uniformly distributed noise over a zero-valued image or simply an image with intensity $1/2$ in every pixel.

Lemma 1 *The moment values of an image with intensity $1/2$ are the half of the moment values of an image with intensity 1.*

Proof The “full-intensity” image $f(x, y) = 1, \forall x, y$, has moment values $mf_{pq} = \sum_{x=0}^{N_1-1} \sum_{y=0}^{N_2-1} x^p y^q f(x, y)$, and exploiting the rectangular form of the image, $mf_{pq} = \sum_{x=0}^{N_1-1} x^p \sum_{y=0}^{N_2-1} y^q$. The image $g(x, y) = 1/2, \forall x, y$ has moment values $mg_{pq} = \sum_{x=0}^{N_1-1} \sum_{y=0}^{N_2-1} x^p y^q g(x, y)$, and exploiting the rectangular form of the image, $mg_{pq} = \frac{1}{2} \sum_{x=0}^{N_1-1} x^p \sum_{y=0}^{N_2-1} y^q = \frac{mf_{pq}}{2}$. \square

Lemma 2 *The moment values of a chessboard image equals approximately the half of the moment values of an image with intensity 1.*

Proof Consider two chessboard images $c_1(x, y)$ and $c_2(x, y)$, which are defined by: $c_1(x, y) = \{1, \text{ if } (x + y) \text{ is even}; 0, \text{ otherwise}\}$, $c_2(x, y) = \{1, \text{ if } (x + y) \text{ is odd}; 0, \text{ otherwise}\}$. The two images complement each other and their sum equals 1, i.e., the full-intensity image: $c_1 + c_2 = 1 = f$. Taking into account the fact that the quantities y^q and $(y + 1)^q$ are close enough for large y and small q values, then it is concluded that $mc_1 = mc_2$. Since $mc_1 + mc_2 = mf$, it is concluded that $mc_1 = mc_2 = mf/2$. \square

Therefore, instead of trying to synthesize the ideal “half-intensity” image, it is adequate to use its moment values, which should be equal to half of the moments of an image with the same size and every pixel value equal to the object level. So, in the remainder of this study, we refer to the “half-intensity” image as an image with moment values equal to half of the moments of full-intensity image.

The substitution of the moments of the m omitted binary images with the moments of the “half-intensity” image results in the approximated moment values m'_{pq} :

$$\begin{aligned} m'_{pq} &= \sum_{i=m}^{n-1} 2^i m b_{ipq} + \sum_{i=0}^{m-1} 2^i m h_{pq} \\ &= \sum_{i=m}^{n-1} 2^i m b_{ipq} + m h_{pq} \sum_{i=0}^{m-1} 2^i \end{aligned} \quad (9)$$

where $m h_{pq}$ are the moments of the “half-intensity” image, i.e., half the values of the moments of the full-intensity image.

The percentage error of the approximation between the exact and the approximated moment values is expressed as:

$$\varepsilon_{pq} = \left| \frac{m_{pq} - m'_{pq}}{m_{pq}} \right| \times 100. \quad (10)$$

The approximation error in the case where some bitplanes are omitted is much bigger in comparison with the case of substitution of the omitted bitplanes with the “half-intensity” image. This is due to the fact that the contribution of the lower-order bitplanes to the moments of the gray image is quite similar to the contribution of the “half-intensity” image to the moments of the gray image. Therefore, the second observation may be considered as experimentally proven.

The percentage contribution Cb_i of the bitplane of the i th plane and the percentage contribution Ch_i of the “half-intensity image” at the i th plane, to the moments of the gray image, are calculated according to:

$$Cb_i = \frac{2^i m b_{ipq}}{m_{pq}} \times 100 \quad (11.1)$$

$$Ch_i = \frac{2^i m h_{pq}}{m_{pq}} \times 100. \quad (11.2)$$

As implied from (11.2), $Ch_i = 2Ch_{i-1}$.

The approximate method is particularly suitable for applications using low-order moments because the approximation error tends to increase for high-moment order.

4.2 Other moment sets

The proposed method is also suitable for the computation of other sets of moments. Specifically, the central moments, the normalized central moments and the Hu’s moment invariants, which are directly related to the geometric moments, may also be computed using the proposed method.

The central moments of an image $f(x, y)$ are invariant under image translation and defined as

$$\mu_{pq} = \sum_{x=0}^{N_1-1} \sum_{y=0}^{N_2-1} (x - \bar{x})^p (y - \bar{y})^q f(x, y) \quad (12)$$

where $\bar{x} = m_{10}/m_{00}$ and $\bar{y} = m_{01}/m_{00}$ are the coordinates of the centroid. Since all image pixels with level 1 belong to the k image blocks and taking into account the rectangular form that appeared within the blocks, (12) may be rewritten as

$$\begin{aligned} \mu_{pq} &= \sum_{i=0}^{k-1} \mu_{pq}^{b_i} = \sum_{i=0}^{k-1} \sum_{x=x_{1,b_i}}^{x_{2,b_i}} \sum_{y=y_{1,b_i}}^{y_{2,b_i}} (x - \bar{x})^p (y - \bar{y})^q \\ &= \sum_{i=0}^{k-1} \left(\sum_{x=x_{1,b_i}}^{x_{2,b_i}} (x - \bar{x})^p \right) \left(\sum_{y=y_{1,b_i}}^{y_{2,b_i}} (y - \bar{y})^q \right) \end{aligned} \quad (13)$$

where $x_{1,b_i}, x_{2,b_i}, y_{1,b_i}, y_{2,b_i}$ are the coordinates of the blocks. The coordinates of the centroid in (13) refer to the center of gravity of the whole image and not to the centroid of each block. The computation of the geometrical moments m_{00}, m_{10}, m_{01} using the proposed methods ensures the fast computation of the centroid of the image. Note that using (13), the complexity is reduced to $O(N)$ instead of $O(N^2)$ using (12).

According to the Appendix, the factor $\sum (x - \bar{x})^p$ can be computed using the analytical formulae

$$\begin{aligned} \sum_{x=x_{1b}}^{x_{2b}} (x - \bar{x})^p &= S_{x_{1b}, x_{2b}}^p - \bar{x} \binom{p}{1} S_{x_{1b}, x_{2b}}^{p-1} + \dots + (-1)^p \\ &\times (x_{2b} - x_{1b} + 1) \bar{x}^p, \quad \forall p \in Z^+ \end{aligned} \quad (14)$$

where S_{ab}^p is defined in (5). The summation of $(y - \bar{y})^q$ in (13) is computed similarly and the fast computation of the central moments is ensured using the above analytical formulae (14). It is clear that Flusser’s proposed refinement and (6) can be also used for the computation of the factors S_{ab}^p .

The normalized central moments of an image are defined as $\eta_{pq} = \frac{\mu_{pq}}{\mu_{00}^\gamma}$ where $\gamma = (p + q)/2 + 1$, $p + q = 2, 3, \dots$ and μ_{pq} are the corresponding central moments of the image. The central moments are required for the computation of the normalized central moments.

Hu’s moment invariants [1] are derived from the normalized central moments. Therefore, the fast computation of the central moments insures the fast computation of the normalized central moments and of the moment invariants.

5 Experimental results

In this section, experimental results for the computation of moments on gray images are presented. Specifically, the determination of the optimal number of bitplanes and “half-intensity” images, the number of the extracted blocks and the computation times are presented here. Also, an analysis of computational complexity is given in this

section. Finally, comparisons with other fast moment computation methods are presented.

All computations are performed using the C programming language and a 1.5 GHz Athlon PC with 1 GB RAM.

5.1 Determination of the optimal number of higher-order bitplanes

Let us consider the substitution of the lower-order binary images with the “half-intensity” image. A large number of indoor and outdoor test images have been used in experiments, to determine the optimum number of binary images that have to be used in moment computation for keeping the approximation error low. In Fig. 1, some of the test images are displayed.

We calculate first the percentage contribution Cb_i , $i = 0, 1, \dots, 7$ of each binary image in the formation of gray moments. Similarly, we calculate the percentage contribution Ch_i , $i = 0, 1, \dots, 7$, of the corresponding bitplanes, when they are replaced by the “half-intensity” image. Table 1 shows these percentage contributions Cb_i , Ch_i for $i = 0, 1, \dots, 7$, for the test image of Fig. 1h. It can be observed that for the first five lower-order planes, both percentage contributions are quite close. For the higher-order bitplanes, the differences in the contributions become larger. Figure 3 illustrates the values of Cb_i , Ch_i for $i = 0, 1, \dots, 5$ in a graphical way. The contribution values are similar for all the sets of test images.

The percentage approximation moment computation error, for $m = 4, 5, 6$, is given in Tables 2, 3 and 4, respectively, where m is the number of “half-intensity” images that were used for the computation of moments for the image in Fig. 1h. As indicated from Tables 2, 3 and 4, the percentage error is quite low for $m = 4$, still at accepted levels for $m = 5$ and increases with m . Note also that the approximation error becomes larger as the moment order increases.

The approximation error for all sets of test images is similar. Specifically, the maximum error for $m = 4$ is 0.57% for moment m_{33} of the image in Fig. 1a. The maximum error for $m = 5$ is 2.32% for moment m_{33} of the image in Fig. 1a. The maximum error for $m = 6$ is 4.14% for moment m_{00} of the image in Fig. 1g.

For pattern analysis and recognition applications, an error less than 3% is usually acceptable. Therefore, from the computation of the percentage errors for the test images and for accepted moment approximation error less than 3%, the optimum number of “half-intensity” images, m , is determined to be $m = 5$. If more accuracy is needed for an application, then the parameter m may be set to a smaller value. In extreme case when lossless moment values are required, the parameter m should be set to the value of 0. In this case of exact moment computation, all the bitplanes of

Table 1 The percentage contributions Cb_i , Ch_i of the binary images b_i and of the “half-intensity” image h , at different plane positions, to the moments of the gray image of Fig. 1h

	Cb_0	Cb_1	Cb_2	Cb_3	Cb_4	Cb_5	Cb_6	Cb_7	Ch_0	Ch_1	Ch_2	Ch_3	Ch_4	Ch_5	Ch_6	Ch_7
m_{00}	0.4035	0.4044	0.8076	0.8087	1.6026	1.6175	3.2155	3.2349	6.4699	6.4956	12.7042	12.9397	22.4905	25.8794	51.8805	51.7589
m_{01}	0.3858	0.3886	0.7771	0.7771	1.5304	1.5543	3.1364	3.1086	6.2172	7.1211	10.5664	12.4344	19.8361	24.8687	56.6467	49.7374
m_{02}	0.3778	0.3817	0.7620	0.7633	1.4990	1.5267	3.1137	3.0534	6.1068	7.2752	9.5462	12.2135	18.9266	24.4271	58.4996	48.8541
m_{03}	0.3752	0.3795	0.7561	0.7591	1.4879	1.5181	3.1148	3.0362	6.0725	7.4343	9.0647	12.1449	18.7916	24.2898	58.9753	48.5796
m_{10}	0.4163	0.4173	0.8351	0.8347	1.6566	1.6694	3.3012	3.3387	6.6775	7.0232	12.7210	13.3550	23.3356	26.7100	50.7110	53.4199
m_{11}	0.3901	0.3933	0.7900	0.7866	1.5515	1.5731	3.1350	3.1463	6.2925	6.8390	10.0361	12.5851	21.6775	25.1701	55.5808	50.3402
m_{12}	0.3766	0.3810	0.7650	0.7621	1.5000	1.5241	3.0439	3.0483	6.0965	6.7275	8.7415	12.1931	21.5391	24.3861	57.3063	48.7722
m_{13}	0.3705	0.3755	0.7532	0.7510	1.4775	1.5019	2.9974	3.0039	6.0077	6.7185	8.0878	12.0154	22.0300	24.0309	57.5651	48.0618
m_{20}	0.4237	0.4244	0.8502	0.8488	1.6794	1.6976	3.3621	3.3952	6.7904	7.1036	12.6997	13.5807	23.8917	27.1615	49.9896	54.3230
m_{21}	0.3945	0.3976	0.8004	0.7952	1.5646	1.5904	3.1674	3.1807	6.3615	6.7039	9.9214	12.7230	23.4793	25.4460	53.9686	50.8920
m_{22}	0.3794	0.3841	0.7729	0.7683	1.5107	1.5365	3.0589	3.0731	6.1462	6.4954	8.6736	12.2923	24.1169	24.5847	54.9921	49.1693
m_{23}	0.3729	0.3785	0.7609	0.7571	1.4904	1.5141	3.0040	3.0282	6.4439	6.0565	8.1125	12.1130	25.2209	24.2260	54.5945	48.4519
m_{30}	0.4279	0.4284	0.8588	0.8567	1.6902	1.7134	3.4027	3.4269	6.8537	7.1444	12.7319	13.7075	24.3006	27.4150	49.4435	54.8299
m_{31}	0.3980	0.4011	0.8084	0.8021	1.5734	1.6043	3.2054	3.2085	6.6277	6.4171	9.9650	12.8341	24.8694	25.6683	52.5526	51.3366
m_{32}	0.3831	0.3881	0.7818	0.7762	1.5231	1.5524	3.0987	3.1048	6.3884	6.2096	8.8226	12.4193	26.1117	24.8385	52.8906	49.6771
m_{33}	0.3775	0.3837	0.7720	0.7673	1.5090	1.5347	3.0499	3.0694	6.1388	6.3404	8.3913	12.2776	27.7033	24.5551	51.8566	49.1102

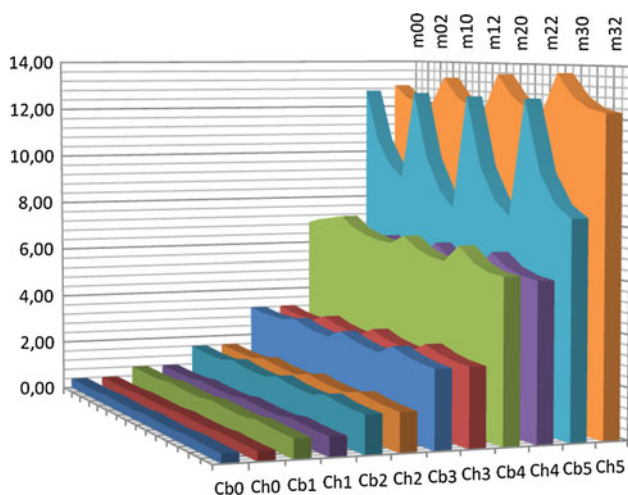


Fig. 3 The percentage contributions of the binary images Cb_i and of the “half-intensity” image at different plane positions Ch_i to the moments of the gray image

Table 2 The percentage approximation moment computation error for $m = 4$ for the image in Fig. 1h

ϵ_{pq}	$p = 0$	$p = 1$	$p = 2$	$p = 3$
$q = 0$	0.0363	0.0510	0.0506	0.0458
$q = 1$	0.0011	0.0326	0.0371	0.0308
$q = 2$	0.0273	0.0299	0.0401	0.0348
$q = 3$	0.0411	0.0337	0.0497	0.0467

Table 3 The percentage approximation moment computation error for $m = 5$ for the image in Fig. 1h

ϵ_{pq}	$p = 0$	$p = 1$	$p = 2$	$p = 3$
$q = 0$	0.3894	0.2948	0.2626	0.2448
$q = 1$	0.9049	0.5139	0.3054	0.1798
$q = 2$	1.1957	0.6011	0.3092	0.1439
$q = 3$	1.4030	0.6771	0.3377	0.1550

Table 4 The percentage approximation moment computation error for $m = 6$ for the image in Fig. 1h

ϵ_{pq}	$p = 0$	$p = 1$	$p = 2$	$p = 3$
$q = 0$	0.1539	0.3392	0.6185	0.7307
$q = 1$	0.9630	2.0351	2.4962	2.6893
$q = 2$	1.4716	2.8505	3.3095	3.4528
$q = 3$	1.6772	3.2505	3.6628	3.7313

the image are used in the calculation using (8). On the other hand, if more speed is needed for an application, then the parameter m may be set to a value of 6.

Table 5 Block number and required times in second for the computation of exact moments up to the order (3,3)

Image	Blocks-8p	IBR-8p	BMC-8p	IMC	Speedup
Fig. 1a	61,121	0.088	0.068	0.481	7.07
Fig. 1b	57,607	0.081	0.070	0.474	6.77
Fig. 1c	59,068	0.086	0.066	0.470	7.12
Fig. 1d	57,399	0.086	0.065	0.473	7.28
Fig. 1e	58,296	0.085	0.069	0.506	7.33
Fig. 1f	75,587	0.098	0.070	0.526	7.51
Fig. 1g	42,406	0.052	0.047	0.315	6.70
Fig. 1h	44,572	0.051	0.049	0.293	5.98
Fig. 1i	43,832	0.055	0.048	0.281	5.85

5.2 Performance evaluation

Let us first consider the case of exact moment computation, where the parameter $m = 0$ and all the bitplanes of the gray image are used in the computation. For the test images of Fig. 1, the required times for the computation of moments up to the order (3,3), with the proposed block-based method using (8) and with the pixel-based computation using (2), are illustrated in Table 5. Column blocks-8p shows the total number of blocks of all the eight bitplanes. Column IBR-8p shows the time for the IBR. Column BMC-8p, shows the time for the computation of the moments up to the order (3,3) with the proposed block-based method, using (8). Column IMC shows the time for the computation of the moments up to the order (3,3) with the pixel-based method, using (2). Comparing the computation of the moments using the proposed block-based method (column BMC-8p) with the conventional pixel-based method (column IMC), it is observed that the proposed method offers a significant computational gain, while providing the exact moment values. The last column (speedup) demonstrates this gain. The average speedup value for all the set of test images of Fig. 1 is 6.85.

Let us examine the case of the substitution of the lower-order bitplanes with the “half-intensity” image. To have significant computational savings, the moments of the “half-intensity” image are stored and their use requires only one addition, as indicated by (9). It is worth noting that at lower bitplanes, the number of the blocks is much larger, and therefore the saving in time from the omission of a lower bitplane is greater than the saving from the omission of a higher bitplane.

As determined in the previous subsection, the value of the parameter m is set to $m = 5$. The required times for the moments computation up to the order (3,3) of the test images of Fig. 1 are presented in Table 6. Column Blocks-3p shows the number of the blocks of the three

Table 6 Blocks number and required times in seconds for the computation of approximated moments up to the order (3,3)

Image	Blocks-3p	IBR-3p	BMC-3p	IMC	Speedup
Fig. 1a	9,825	0.012	0.011	0.481	43.73
Fig. 1b	5,498	0.007	0.006	0.474	79.00
Fig. 1c	5,986	0.008	0.006	0.470	78.33
Fig. 1d	6,121	0.009	0.007	0.473	67.57
Fig. 1e	6,504	0.008	0.007	0.506	72.29
Fig. 1f	12,508	0.014	0.014	0.526	37.57
Fig. 1g	5,925	0.006	0.006	0.315	52.50
Fig. 1h	6,538	0.007	0.007	0.293	41.86
Fig. 1i	5,468	0.005	0.007	0.281	40.14

higher-order bitplanes. Column IBR-3p presents the required time for the IBR process for the three higher bitplanes. Column BMC-3p presents the required time for the computation of moments using the proposed method, while column IMC presents the required time for the computation of moments using the pixel-based method (2). The last column presents the speedup value obtained using the proposed method 2, instead of the pixel-based computation. The average speedup value for all sets of test images is 57.00.

Note that the whole time of BMC with IBR is less than 0.033 s, and therefore the proposed method operates in real time with more than 30 images/s.

5.3 Computational complexity

It is clear that the computational complexity of the proposed method is proportional to the number k of the blocks and the number L^2 of the moments of order up to $(L - 1, L - 1)$ and, therefore, the complexity of the proposed method is $O(kL^2)$. A direct analysis of the computational complexity cannot be done, since k is image dependent. However, as seen from Table 6, the number of blocks k of the three higher binary images is quite low and $k \ll N^2$. Moreover, the use of precalculated moment values for the “half-intensity” image and the precalculation of the terms $(x + 0.5)^{p+1}/p + 1$ in (6) permit the real-time implementation in a low-cost serial computer.

5.4 Comparisons

In this subsection, comparisons of the proposed method with that of Chung [27] and of Hosny [28] are provided.

Chung and Chen method [27], already mentioned in the introduction, extended the idea of [22] to gray images by separating gray images to “homogeneous” blocks of image intensities and performing moment calculation

using the blocks. The definition of homogeneity introduces an error parameter ϵ , which controls the process of forming the blocks. In a specified block, its pixel intensities are estimated using 2D linear interpolation based on the corner intensities of the block. Then, this block is considered homogenous if each original image pixel intensity of the block has an absolute error less than ϵ with respect to the corresponding estimated pixel intensity. Otherwise, the block is subdivided into two equally sized sub-blocks and the procedure is repeated. In the sequel, the moments of each homogenous block and consequently of the entire image are computed using the techniques in [22]. Apparently, the choice of ϵ influences the moment calculation error making it larger for large values of ϵ . Due to (2) even small differences in image intensities may have a serious influence on the accuracy of the moment calculation. On the other hand, large values of ϵ result in a small number of homogenous blocks.

Since it is difficult to associate Chung’s parameter ϵ with a moment approximation error less than 3%, we have chosen to compare both methods when they provide exact moment values (lossless operation). In Chung’s method, this happens when $\epsilon = 0$. Then the block representation is lossless and the exact reconstruction of the gray image from its blocks is possible. Also, the computation of the moments is error free. In the proposed method, the lossless operation is achieved when $m = 0$.

Hosny’s method [28], as mentioned in the introduction, exploits the rectangular form of the pixel to separate the 2D computation in two steps, by successive computation of 1D moments for each image row, and provides exact moment values. Since Hosny’s method provides exact moment values, the comparison is with our proposed method for $m = 0$.

Table 7 illustrates the required time for the computation of moments up to the order (3,3) for all the test images of

Table 7 The required times in second for the exact computation of the moments up to the order (3,3) using different methods

Image	BMC-8p	CHUNG	HOSNY
Fig. 1a	0.068	0.126	0.049
Fig. 1b	0.070	0.124	0.051
Fig. 1c	0.066	0.124	0.048
Fig. 1d	0.065	0.126	0.052
Fig. 1e	0.069	0.130	0.052
Fig. 1f	0.070	0.134	0.046
Fig. 1g	0.047	0.074	0.031
Fig. 1h	0.049	0.076	0.033
Fig. 1i	0.048	0.080	0.033

Fig. 1. Column BMC-8p presents the required time using eight bitplanes and the exact moments' computation. Column CHUNG presents the required time with Chung's method. Column HOSNY presents the required time for the computation using the method of Hosny. It is apparent that Hosny's method outperforms the other two methods, second is our proposed method and third is the method of Chung. However, as shown in Table 6, the use of our proposed method for $m = 5$ outperforms the method of Hosny.

6 Conclusion

In this work, a real-time moment computation method on gray images is proposed. The proposed method is based on a combination of a previous work and some new considerations. The previous work is the representation of a binary image with blocks and the real-time computation of moments using the blocks. The new work is based on the observation that the lower-order bitplanes are similar to a so-called "half-intensity" image, which has moment values equal to the half of the moment values of the full-intensity image. It has been experimentally proved that the moments of the lower-order bitplanes are quite similar to the moments of the "half-intensity" image. Therefore, the lower bitplanes are omitted from the calculation and the precalculated moment values of the "half-intensity" image are used instead. This results in significant computational saving while preserving a significant degree of accuracy in moment computation.

The idea is simple, but very effective. The combination with a previous work results in the fastest method presently for moment computation, which can be implemented in real time even with software implementation in low-cost serial computers.

Appendix

Computation of the factor $\sum_{x=x_{1b}}^{x_{2b}} (x - \bar{x})^p$ in central moments

Using the known identities:

$$\begin{aligned}
 (c - d)^2 &= c^2 - 2cd + d^2 \\
 (c - d)^3 &= c^3 - 3c^2d + 3cd^2 - d^3 \\
 (c - d)^4 &= c^4 - 4c^3d + 6c^2d^2 - 4cd^3 + d^4 \\
 (c - d)^m &= c^m - \binom{m}{1}c^{m-1}d + \binom{m}{2}c^{m-2}d^2 - \dots + (-1)^m d^m
 \end{aligned}
 \tag{15}$$

it is concluded that

$$\begin{aligned}
 \sum_{x=1}^{x_{2b}} (x - \bar{x}) &= S_{1,x_{2b}}^1 - x_{2b}\bar{x}, \\
 \sum_{x=1}^{x_{2b}} (x - \bar{x})^2 &= S_{1,x_{2b}}^2 - 2\bar{x}S_{1,x_{2b}}^1 + x_{2b}\bar{x}^2 \\
 \sum_{x=1}^{x_{2b}} (x - \bar{x})^3 &= S_{1,x_{2b}}^3 - 3\bar{x}S_{1,x_{2b}}^2 + 3\bar{x}^2S_{1,x_{2b}}^1 - x_{2b}\bar{x}^3, \\
 \sum_{x=1}^{x_{2b}} (x - \bar{x})^4 &= S_{1,x_{2b}}^4 - 4\bar{x}S_{1,x_{2b}}^3 + 6\bar{x}^2S_{1,x_{2b}}^2 - 4\bar{x}^3S_{1,x_{2b}}^1 + x_{2b}\bar{x}^4 \\
 \sum_{i=1}^{x_{2b}} (x - \bar{x})^p &= S_{1,x_{2b}}^p - \bar{x} \binom{p}{1} S_{1,x_{2b}}^{p-1} + \dots + (-1)^p x_{2b} \bar{x}^p, \quad \forall p \in Z^+
 \end{aligned}
 \tag{16}$$

and

$$\begin{aligned}
 \sum_{x=x_{1b}}^{x_{2b}} (x - \bar{x}) &= \sum_{x=1}^{x_{2b}} (x - \bar{x}) - \sum_{x=1}^{x_{1b}-1} (x - \bar{x}) \\
 &= S_{x_{1b},x_{2b}}^1 - (x_{2b} - x_{1b} + 1)\bar{x} \\
 \sum_{x=x_{1b}}^{x_{2b}} (x - \bar{x})^2 &= \sum_{x=1}^{x_{2b}} (x - \bar{x})^2 - \sum_{x=1}^{x_{1b}-1} (x - \bar{x})^2 = S_{x_{1b},x_{2b}}^2 \\
 &\quad - 2\bar{x}S_{x_{1b},x_{2b}}^1 + (x_{2b} - x_{1b} + 1)\bar{x}^2 \\
 \sum_{x=x_{1b}}^{x_{2b}} (x - \bar{x})^3 &= \sum_{x=1}^{x_{2b}} (x - \bar{x})^3 - \sum_{x=1}^{x_{1b}-1} (x - \bar{x})^3 = S_{x_{1b},x_{2b}}^3 \\
 &\quad - 3\bar{x}S_{x_{1b},x_{2b}}^2 + 3\bar{x}^2S_{x_{1b},x_{2b}}^1 - (x_{2b} - x_{1b} + 1)\bar{x}^3 \\
 \sum_{x=x_{1b}}^{x_{2b}} (x - \bar{x})^4 &= \sum_{x=1}^{x_{2b}} (x - \bar{x})^4 - \sum_{x=1}^{x_{1b}-1} (x - \bar{x})^4 \\
 &= S_{x_{1b},x_{2b}}^4 - 4\bar{x}S_{x_{1b},x_{2b}}^3 + 6\bar{x}^2S_{x_{1b},x_{2b}}^2 - 4\bar{x}^3S_{x_{1b},x_{2b}}^1 \\
 &\quad + (x_{2b} - x_{1b} + 1)\bar{x}^4 \\
 \sum_{x=x_{1b}}^{x_{2b}} (x - \bar{x})^p &= S_{x_{1b},x_{2b}}^p - \bar{x} \binom{p}{1} S_{x_{1b},x_{2b}}^{p-1} + \dots + (-1)^p \\
 &\quad \times (x_{2b} - x_{1b} + 1)\bar{x}^p, \quad \forall p \in Z^+
 \end{aligned}
 \tag{17}$$

where $S_{a,b}^p$ have been defined in (5).

References

- Hu, M.K.: Visual pattern recognition by moment invariants. IRE Trans. Inf. Theory **8**, 179–187 (1962)
- Belkasim, S.O., Shridhar, M., Ahmadi, M.: Pattern recognition with moment invariants: a comparative study and new results. Pattern Recognit. **24**, 1117–1138 (1991)
- Prokop, R.J., Reeves, A.P.: A survey of moment-based techniques for unoccluded object representation and recognition. Graph. Models Image Process. **54**(5), 438–460 (1992)
- Sluzek, A.: Identification and inspection of 2-D objects using new moment-based shape descriptors. Pattern Recognit. Lett. **16**(12), 687–697 (1995)
- Flusser, J., Suk, T.: Rotation moment invariants for recognition of symmetric objects. IEEE Trans. Image Process. **15**(12), 3784–3790 (2006)

6. Teague, M.R.: Image analysis via the general theory of moments. *J. Opt. Soc. Am.* **70**, 920–930 (1980)
7. Abu-Mostafa, Y.S., Psaltis, D.: Recognitive aspects of moment invariants. *IEEE Trans. Pattern Anal. Mach. Intell.* **6**(6), 698–706 (1984)
8. Teh, C.-H., Chin, R.T.: On image analysis by the method of moments. *IEEE Trans. Pattern Anal. Mach. Intell.* **10**(4), 496–513 (1988)
9. Khotanzad, A., Hong, Y.H.: Invariant image recognition by Zernike moments. *IEEE Trans. Pattern Anal. Mach. Intell.* **12**(5), 489–497 (1990)
10. Zhu, Y., De Silva, L.C., Ko, C.C.: Using moment invariants and HMM in facial expression recognition. *Pattern Recognit. Lett.* **23**, 83–91 (2002)
11. Bhattacharya, D., Sinha, S.: Invariance of stereo images via the theory of complex moments. *Pattern Recognit.* **30**(9), 1373–1386 (1997)
12. Chueng, K.K.T., Ip, H.S.: Image Retrieval in Digital Library based on Symmetry Detection. In: *Proceedings of the Computer Graphics International*, pp. 366–372. Hannover, Germany (1998)
13. Xin, Y., Liao, S., Pawlak, M.: Circularly orthogonal moments for geometrically robust image watermarking. *Pattern Recognit.* **40**(12), 3740–3752 (2007)
14. Yang, L., Albregtsen, F.: Fast and exact computation of Cartesian geometric moments using discrete Green's theorem. *Pattern Recognit.* **29**(7), 1061–1073 (1996)
15. Liao, S.X., Pawlak, M.: On image analysis by moments. *IEEE Trans. Pattern Anal. Mach. Intell.* **18**(3), 254–266 (1996)
16. Li, B.C., Shen, J.: Fast computation of moment invariants. *Pattern Recognit.* **24**(8), 807–813 (1991). doi:[10.1016/0031-3203\(91\)90048-A](https://doi.org/10.1016/0031-3203(91)90048-A)
17. Jiang, X.Y., Bunke, H.: Simple and fast computation of moments. *Pattern Recognit.* **24**(8), 801–806 (1991)
18. Zakaria, M.F., Vroomen, L.J., Zsombor-Murray, P.J.A., van Kessel, J.M.H.M.: Fast algorithm for the computation of moment invariants. *Pattern Recognit.* **20**(6), 639–643 (1987)
19. Dai, M., Baylou, P., Najim, M.: An efficient algorithm for computation of shape moments from run-length codes or chain codes. *Pattern Recognit.* **25**(10), 1119–1128 (1992)
20. Li, B.C.: A new computation of geometric moments. *Pattern Recognit.* **26**(1), 109–113 (1993)
21. Flusser, J.: Fast calculation of geometric moments of binary images. In: *Proceedings of 22nd OAGM'98 Workshop Pattern Recognition. Medical Computer Vision*, pp. 265–274, Illmitz, Austria (1998)
22. Spiliotis, I.M., Mertzios, B.G.: Real-time computation of two-dimensional moments on binary images using image block representation. *IEEE Trans. Image Process.* **7**(11), 1609–1615 (1998)
23. Spiliotis, I.M., Mertzios, B.G.: Fast algorithms for basic processing and analysis operations on block-represented binary images. *Pattern Recognit. Lett.* **17**(14), 1437–1450 (1996)
24. Spiliotis, I.M., Mertzios, B.G.: A fast parallel skeletonization algorithm on block-represented binary images. *Elektrik* **1**(1), 161–173 (1997)
25. Spiliotis, I.M., Mertzios, B.G.: A fast skeleton algorithm on block-represented binary images. In: *Proceedings of 13th International Conference on Digital Signal Processing (DSP97)*, Santorini, Hellas (1997)
26. Perantonis, S.J., Gatos, B., Papamarkos, N.: Block decomposition and segmentation for fast Hough transform evaluation. *Pattern Recognit.* **32**(5), 811–824 (1999)
27. Chung, K.L., Chen, P.-C.: An efficient algorithm for computing moments on a block representation of a grey-scale image. *Pattern Recognit.* **38**(12), 2578–2586 (2005)
28. Hosny, K.M.: Exact and fast computation of geometric moments for gray-level images. *Appl. Math. Comput.* **189**, 1214–1222 (2007)
29. Spiliotis, I.M., Boutalis, Y. (2008) Fast and real-time moment computation methods of gray images using image block representation. In: *Proceedings of 5th IASTED International Conference on Signal Processing, Pattern Recognition and Applications (SPPRA-2008)*, pp. 323–328. Innsbruck, Austria (2008)
30. Flusser, J.: Refined moments calculation using image block representation. *IEEE Trans. Image Process.* **9**(11), 1977–1978 (2000)

Author Biographies

Dr. Iraklis Spiliotis received the Diploma Degree and the PhD Degree in Electrical Engineering from the Democritus University of Thrace, Hellas. He is currently with the Laboratory of Digital Systems, Electrical and Computer Engineering Department, Democritus University of Thrace, Hellas. His research interests include algorithms for signal and image processing, pattern recognition, image retrieval and reasoning.



Dr. Yiannis Boutalis (M'86) received the diploma of Electrical Engineer in 1983 from Democritus University of Thrace (DUTH), Greece and the PhD degree in Electrical and Computer Engineering (topic Image Processing) in 1988 from the Computer Science Division of National Technical University of Athens, Greece. Since 1996, he serves as a faculty member, at the Department of Electrical and Computer Engineering, DUTH, Greece, where he is currently an Associate Professor and director of the Automatic Control Systems lab. Currently, he is also a Visiting Professor for research cooperation at Friedrich-Alexander University of Erlangen-Nuremberg, Germany, chair of Automatic Control. He served as an assistant visiting professor at University of Thessaly, Greece, and as a visiting professor in Air Defence Academy of General Staff of airforces of Greece. He also served as a researcher in the Institute of Language and Speech Processing (ILSP), Greece, and as a managing director of the R&D SME Ideatech S.A, Greece, specializing in pattern recognition and signal processing applications. His current research interests are focused in the development of Computational Intelligence techniques with applications in Control, Pattern Recognition, Signal and Image Processing Problems.