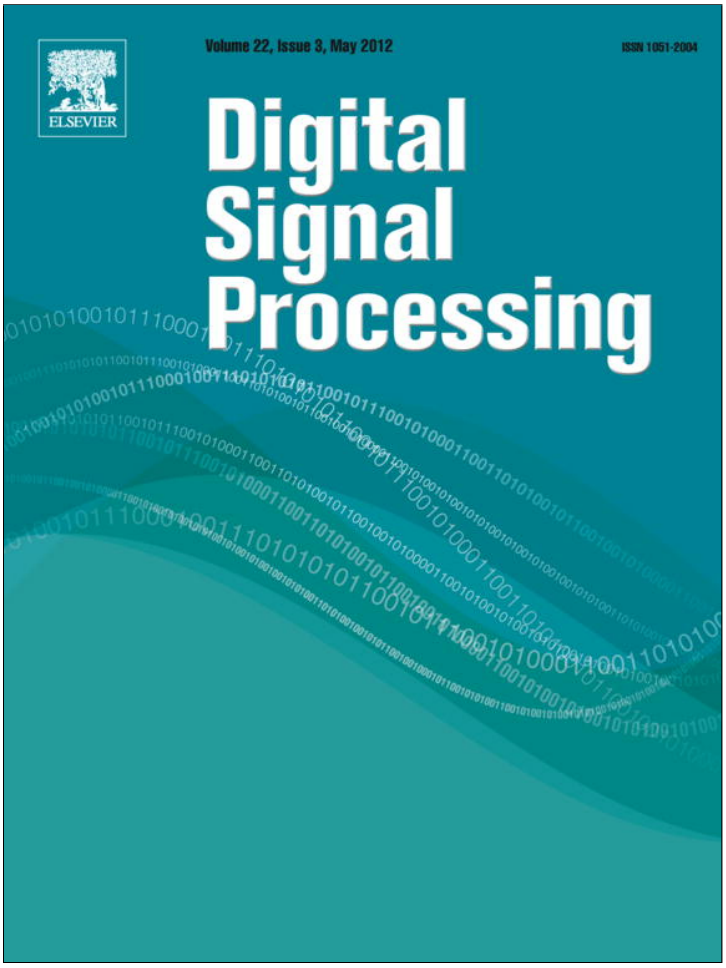
Volume 22, Issue 3, May 2012
ISSN 1051-2004
ELSEVIER
Digital
Signal
Processing

Digital Signal Processing 22 (2012) 476–485

Contents lists available at SciVerse ScienceDirect

Digital Signal Processing

www.elsevier.com/locate/dsp

Fast computation of accurate Gaussian–Hermite moments for image processing applications

Khalid M. Hosny

Department of Information Technology, Faculty of Computers and Informatics, Zagazig University, Zagazig, Egypt

ARTICLE INFO

Article history:
Available online 12 January 2012

Keywords:
Gaussian–Hermite moments
Accurate computation
Fast algorithm
Gray-level images
Noise robustness

ABSTRACT

Gaussian–Hermite moments are orthogonal moments widely used in image processing and computer vision applications. Similar to the other families of orthogonal moments, highly computational demands represent the main challenging. In this work, an efficient method is proposed for fast computation of highly accurate Gaussian–Hermite moments for gray-level images. The proposed method achieves the accuracy through the integration of Gaussian–Hermite polynomials over the image pixels. To achieve the efficiency, the symmetry property of Gaussian–Hermite polynomials is employed where the computational complexity is reduced by 75%. Fast computational methodology is employed to significantly accelerate the computational process where the 2D Gaussian–Hermite moments are treated in a separated form. Numerical experiments are performed where the results are compared with the conventional method. The comparison of the obtained results clearly ensures the efficiency of the proposed method.

1. Introduction

Moments and functions of moments have been widely used in different applications of pattern recognition, image processing and computer vision [1–6]. Geometric moments and their translation, scaling and rotation invariants were firstly introduced and implemented by Hu [7]. In the early 80's of the last century, Teague [8] introduced the concept of orthogonal moments, where continuous orthogonal polynomials are used to generate moments for image analysis. Legendre, Zernike and pseudo Zernike are examples of continuous orthogonal moments. Teh and Chin [9] showed that, orthogonal moments could be used to represent an image with the minimum amount of information redundancy.

Gaussian–Hermite moments represent another kind of continuous orthogonal moments. These moments were firstly introduced by Shen [10]. Shen et al. [11] compared the performance of Gaussian–Hermite moments as orthogonal moments and the geometric moments. Later, Wu and Shen [12] discussed the properties of the orthogonal Gaussian–Hermite moments and their applications. Gaussian–Hermite moments are used in detection of the moving objects [13–15], fingerprint segmentation and classifications [16–18], medical image segmentation [19], stereo matching [20], image denoising [21], iris recognition [22] and license plate character recognition [23]. Recently, Yang and his co-authors [24] derived the rotation and translation invariants of Gaussian–Hermite moments. It is clear that, the computational process of rotation and translation Gaussian–Hermite moment invariants is dependent on the computation of the original Gaussian–Hermite moments.

The conventional computation of continuous orthogonal moments includes numerical approximation which results in by replacing integration by a truncated finite summation. Liao and Pawlak [25] attempted to overcome this problem by using a modified approximation method. Recently, exact computation of moments by integrating their polynomial functions over image pixels is an elegant approach proposed by Hosny for efficient and accurate computation of geometric moments [26], Legendre moments [27], radial moments [28], and Zernike moments [29,30].

This paper proposes a fast method for accurate computation of orthogonal Gaussian–Hermite moments for binary and gray-level images. The 2D Gaussian–Hermite moments are computed by applying the approach of mathematical integration of Gaussian–Hermite functions over digital image pixels where the approximation errors are avoided. The symmetry property of Gaussian–Hermite functions is employed to achieve a significant reduction in the computational process. The conducted numerical experiments clearly show the efficiency of the proposed method.

The rest of the paper is organized as follows: In Section 2, a concise presentation of orthogonal Gaussian–Hermite moments and their approximation computation is given. In Section 3, a detailed description of the proposed method for computation of accurate Gaussian–Hermite moments is presented. Section 4 is devoted to numerical experiments and results. Conclusion is presented in Section 5.

E-mail address: k_hosny@yahoo.com.

1051-2004/$ – see front matter © 2012 Elsevier Inc. All rights reserved.
doi:10.1016/j.dsp.2012.01.002

2. Gaussian–Hermite moments

Hermite polynomials are orthogonal polynomials defined over the domain $(-\infty, \infty)$ where the Hermite polynomial of degree p is defined as follows [31]:

$$H_p(x) = (-1)^p e^{x^2} \frac{d^p}{dx^p}\left(e^{-x^2}\right) \tag{1}$$

Such polynomial could be defined in an explicit expansion as follows:

$$H_p(x) = p! \sum_{m=0}^{\lfloor \frac{p}{2} \rfloor} (-1)^m \frac{1}{m!(p-2m)!} (2x)^{p-2m} \tag{2}$$

where the operator $\lfloor p/2 \rfloor$ is equal to $(p-1)/2$ if p is odd otherwise equal to $p/2$. The recurrence relation of Hermite polynomials is:

$$H_{p+1}(x) = 2xH_p(x) - 2pH_{p-1}(x) \tag{3}$$

where $p \geqslant 1$ and the first two polynomials are $H_0(x) = 1$ and $H_1(x) = 2x$. The orthogonality relation of Hermite polynomials with respect to the weight function, e^{-x^2}, is defined as follows:

$$\int_{-\infty}^{\infty} e^{-x^2} H_p(x) H_q(x)\, dx = 2^p p! \sqrt{\pi}\, \delta_{pq} \tag{4}$$

Based on Eq. (4), the normalized Hermite polynomials are defined using original Hermite polynomials as follows:

$$\hat{H}_p(x) = \frac{1}{\sqrt{2^p p! \sqrt{\pi}}} e^{(-\frac{x^2}{2})} H_p(x) \tag{5}$$

The normalized Hermite polynomials satisfy the following orthogonality property:

$$\int_{-\infty}^{\infty} \hat{H}_p(x) \hat{H}_q(x)\, dx = \delta_{pq} \tag{6}$$

Replacing x by x/σ, Gaussian–Hermite functions are defined using the normalized Hermite function as follows:

$$\hat{H}_p(x/\sigma) = \frac{1}{\sqrt{2^p p! \sigma \sqrt{\pi}}} e^{(-\frac{x^2}{2\sigma^2})} H_p(x/\sigma) \tag{7}$$

The parameter σ is the standard deviation. Gaussian–Hermite moments of order $(p+q)$ for the image intensity function, $f(x, y)$, is defined as:

$$M_{pq} = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} f(x, y) \hat{H}_p(x/\sigma) \hat{H}_q(y/\sigma)\, dx\, dy \tag{8}$$

For a digital image of size $M \times N$, the approximated Gaussian–Hermite moments are computed by using the following formula [32]:

$$\tilde{M}_{pq} = \sum_{i=0}^{M-1} \sum_{j=0}^{N-1} f(x_i, y_j) \hat{H}_p(x_i/\sigma) \hat{H}_q(y_j/\sigma) \Delta x \Delta y \tag{9}$$

In this formula, the double integration in Eq. (8) is replaced by double summations which results in numerical error. Based on the principles of mathematical analysis, summations are equivalent to integrals as the number of sampling points tends to infinity which is impossible in the limited computing environment. The numerical error increased as the number of sampling points decreased. Also, this error increased as the order of moments increased. Therefore, numerical instabilities could be encountered when the moment order reaches a cretin value. The optimum way to overcome this problem is the accurate evaluation of the double integration in Eq. (8).

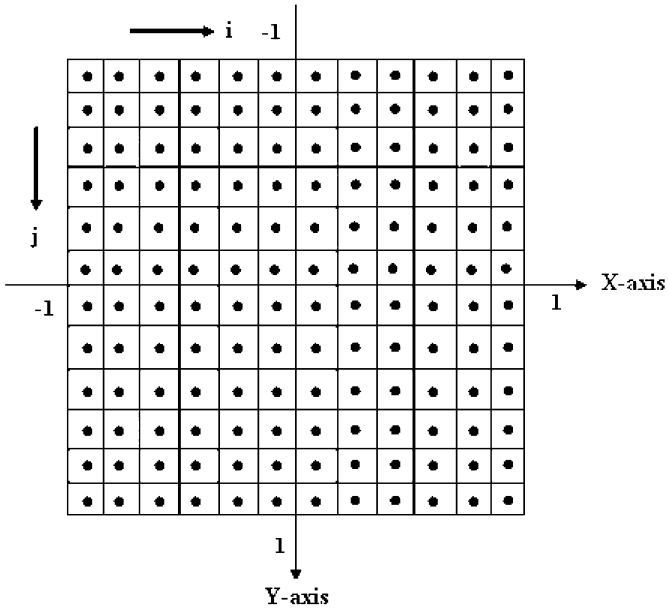

Fig. 1. Input image defined in the square $[-1, 1] \times [-1, 1]$.

3. The proposed method

Direct computation of Gaussian–Hermite moments by using Eq. (9) is impractical where two major challenges were raised. Inaccurate computed moments represent the first challenge while the highly computational costs represent the second challenge. The proposed method aims to overcome these two problems by presenting a fast and exact-like computation of Gaussian–Hermite moments.

To achieve these goals, the input digital image of size $M \times N$ is defined as an array of pixels. Centers of these pixels are the points (x_i, y_j), where the image intensity function is defined only for this discrete set of points $(x_i, y_j) \in [-1, 1] \times [-1, 1]$ as displayed in Fig. 1. $\Delta x_i = x_{i+1} - x_i$, $\Delta y_j = y_{j+1} - y_j$ are sampling intervals in the x- and y-directions respectively. In the literature of digital image processing, the intervals Δx_i and Δy_j are fixed at constant values $\Delta x_i = 2/M$, and $\Delta y_j = 2/N$ respectively. The points (x_i, y_j) are defined where $x_i = -1 + (i - 0.5)\Delta x$; $y_j = -1 + (j - 0.5)\Delta y$; $i = 1, 2, 3, \ldots, M$, and $j = 1, 2, 3, \ldots, N$.

3.1. Accurate Gaussian–Hermite moments

By substituting Eq. (2) in (7), the Gaussian–Hermite functions could be rewritten as follows:

$$\hat{H}_p(x/\sigma) = \frac{1}{\sqrt{2^p p! \sigma \sqrt{\pi}}} e^{(-\frac{x^2}{2\sigma^2})} \sum_{m=0}^{\lfloor \frac{p}{2} \rfloor} (-1)^m \frac{p!}{m!(p-2m)!} (2x)^{p-2m} \tag{10}$$

Rearrange the mathematical terms in the right-hand side, Eq. (10) could be rewritten as follows:

$$\hat{H}_p(x/\sigma) = C_p(\sigma) \sum_{n=0}^{\lfloor \frac{p}{2} \rfloor} B_{p,m} \left(\frac{x}{\sigma}\right)^{p-2m} e^{(-\frac{x^2}{2\sigma^2})} \tag{11}$$

where the coefficients $C_p(\sigma)$ and $B_{p,m}$ are defined as:

$$C_p(\sigma) = \frac{1}{\sqrt{2^p p! \sigma \sqrt{\pi}}} \tag{12}$$

$$B_{p,m} = (-1)^m \frac{p!}{m!(p-2m)!}(2)^{p-2m} \tag{13}$$

It is clear that, both $C_p(\sigma)$ and $B_{p,m}$ are image independents, therefore, the values of these coefficients could be pre-computed, stored and recalled whenever needed to avoid any excessive computational demands. For efficient computation, these coefficients are computed recursively.

Based on the theory of moment computation [25]; Eq. (8) could be rewritten as follows:

$$M_{pq} = C_p(\sigma)C_q(\sigma)\sum_{i=1}^{M}\sum_{j=1}^{N} h_{pq}(x_i, y_j) f(x_i, y_j) \tag{14.1}$$

where

$$h_{pq}(x_i, y_j) = \int_{x_i-\frac{\Delta x_i}{2}}^{x_i+\frac{\Delta x_i}{2}} \int_{y_j-\frac{\Delta y_j}{2}}^{y_j+\frac{\Delta y_j}{2}} \hat{H}_p(x/\sigma)\hat{H}_q(y/\sigma)\,dx\,dy \tag{14.2}$$

Based on the foundations of mathematical analysis, Eq. (14.2) could be rewritten in a separable form as follows:

$$h_{pq}(x_i, y_j) = \left(\int_{x_i-\frac{\Delta x_i}{2}}^{x_i+\frac{\Delta x_i}{2}} \hat{H}_p(x/\sigma)\,dx\right)\left(\int_{y_j-\frac{\Delta y_j}{2}}^{y_j+\frac{\Delta y_j}{2}} \hat{H}_q(y/\sigma)\,dy\right) \tag{15}$$

By using the definition of Gaussian–Hermite function in (11); Eq. (15) is rewritten as follows:

$$h_{pq}(x_i, y_j) = IX_p(x_i) IY_q(y_j) \tag{16}$$

where

$$IX_p(x_i) = \left(C_p(\sigma)\sum_{m=0}^{\lfloor \frac{p}{2} \rfloor} \frac{B_{p,m}}{\sigma^{p-2m}}\left(\int_{x_i-\frac{\Delta x_i}{2}}^{x_i+\frac{\Delta x_i}{2}} x^{p-2m} e^{\left(-\frac{x^2}{2\sigma^2}\right)}\,dx\right)\right) \tag{17.1}$$

$$IY_q(y_j) = \left(C_q(\sigma)\sum_{n=0}^{\lfloor \frac{q}{2} \rfloor} \frac{B_{q,n}}{\sigma^{q-2n}}\left(\int_{y_j-\frac{\Delta y_j}{2}}^{y_j-\frac{\Delta y_j}{2}} y^{q-2n} e^{\left(-\frac{y^2}{2\sigma^2}\right)}\,dy\right)\right) \tag{17.2}$$

Upper and lower limits of the integration in Eqs. (17.1) and (17.2) will be expressed as follows:

$$U_{i+1} = x_i + \frac{\Delta x_i}{2}, \qquad U_i = x_i - \frac{\Delta x_i}{2} \tag{18}$$

Similarly,

$$V_{j+1} = y_j + \frac{\Delta y_j}{2}, \qquad V_j = y_j - \frac{\Delta y_j}{2} \tag{19}$$

Accurate computation of the kernels, $IX_p(x_i)$ and $IY_q(y_j)$, is the key point in accurate computation of Gaussian–Hermite moments. The computation of Eqs. (17.1) and (17.2) includes the computation of the coefficients $C_p(\sigma)$ and $B_{p,m}$, plus the evaluation of the definite integrals. Since the coefficients $C_p(\sigma)$ and $B_{p,m}$ are computed exactly, then, it is concluded that, accurate evaluate of the definite integrals in Eqs. (17.1) and (17.2) is the key point in accurate computation of the kernels $IX_p(x_i)$ and $IY_q(y_j)$. Eqs. (17.1) and (17.2) are rewritten in a compact form as follows:

$$IX_p(x_i) = C_p(\sigma)\sum_{m=0}^{\lfloor \frac{p}{2} \rfloor} \frac{B_{p,m}}{\sigma^{p-2m}} I_{p,m}(i) \tag{20.1}$$

$$IY_q(y_j) = C_q(\sigma)\sum_{n=0}^{\lfloor \frac{q}{2} \rfloor} \frac{B_{q,n}}{\sigma^{q-2n}} I_{q,n}(j) \tag{20.2}$$

where

$$I_{p,m}(i) = \int_{U_i}^{U_{i+1}} x^{p-2m} e^{\left(-\frac{x^2}{2\sigma^2}\right)}\,dx \tag{21.1}$$

$$I_{q,n}(j) = \int_{V_j}^{V_{j+1}} y^{q-2n} e^{\left(-\frac{y^2}{2\sigma^2}\right)}\,dy \tag{21.2}$$

In order to evaluate these definite integrals, the rule of integration by parts is applied. Both integrals could be simply rewritten in the following form:

$$I_T(i) = \int_{U_i}^{U_{i+1}} x^T e^{\left(-\frac{x^2}{2\sigma^2}\right)}\,dx \tag{22}$$

where the index T is a non-negative integer; $T = 0, 1, 2, \ldots, Max$, and *Max* is the maximum order of Gaussian–Hermite moment. The definite integration defined by Eq. (22) must be accurately evaluated for all values of the parameter T. Based on the rule of integration by parts, a series of mathematical operations are performed to derive a formula for accurate evaluation of the definite integration defined in Eq. (22). This formula is expressed as follows:

$$I_T = \sigma^2\left[(U_i)^{T-1} e^{\left(-\frac{U_i^2}{2\sigma^2}\right)} - (U_{i+1})^{T-1} e^{\left(-\frac{U_{i+1}^2}{2\sigma^2}\right)} + (T-1) I_{T-2}\right] \tag{23}$$

where $T \geqslant 2$; I_0 and I_1 are known. The implementation of formula defined by Eq. (23) required the evaluation of the first two definite integrals, I_0 and I_1. The first definite integral, I_0, is defined as follows:

$$I_0 = \int_{U_i}^{U_{i+1}} e^{\left(-\frac{x^2}{2\sigma^2}\right)}\,dx \tag{24}$$

This integration is difficult to be evaluated analytically. So, an accurate numerical integration method could be a good choice. The composite Simpson's rule is proved to be very accurate where the numerical and exact values are almost equal [33]. This rule is represented by using the following equation:

$$\begin{aligned}\int_a^b f(x)\,dx \approx \frac{h}{48}\Bigg\{&17.0\big[f(x_0) + f(x_L)\big] + 59.0\big[f(x_1) + f(x_{L-1})\big]\\ &+ 43.0\big[f(x_2) + f(x_{L-2})\big] + 49.0\big[f(x_3) + f(x_{L-3})\big]\\ &+ 48.0\sum_{t=4}^{L-4} f(x_t)\Bigg\}\end{aligned} \tag{25}$$

Any definite integral of the form $\int_a^b f(x)\,dx$ could be evaluated numerically by using composite Simpson's rule defined in Eq. (25). All details and the required pseudo code for the implementation of this method could be found in [33]. The second required definite integral, I_1, is written as follows:

$$I_1 = \int_{U_i}^{U_{i+1}} x e^{\left(-\frac{x^2}{2\sigma^2}\right)} dx \tag{26}$$

Fortunately, this definite integral could be analytically evaluated by applying the rules of integration by parts. The definite integration I_1 is exactly computed as follows:

$$I_1 = \sigma^2 \left[e^{\left(-\frac{U_i^2}{2\sigma^2}\right)} - e^{\left(-\frac{U_{i+1}^2}{2\sigma^2}\right)} \right] \tag{27}$$

By using Eq. (23) in (17) and (16), the set of orthogonal Gaussian–Hermite moments are computed. It is clear that, the kernels $IX_p(x_i)$ and $IY_q(y_j)$ are image-independent; therefore it could be pre-computed, stored, recalled whenever it is needed to avoid repetitive computation.

3.2. Computational aspects

Efficient computation of orthogonal moments of an image is an important target. Direct computation of Gaussian–Hermite moments is very time-consuming and neither relevant to real time applications nor big size images. For efficient computation process, a fast computational algorithm is proposed. This algorithm consists of two main procedures. In the first procedure, the symmetry property of Gaussian–Hermite polynomials is employed where 75% of the computational complexity is reduced. In the second procedure, a fast algorithm is implemented to accelerate the computational process. A detailed description of these procedures could be found through the following subsections.

3.2.1. Symmetry property

The Gaussian–Hermite polynomials have symmetry properties in both directions [31]. These symmetry properties are defined as follows:

$$\hat{H}_p(-x/\sigma) = (-1)^p \hat{H}_p(x/\sigma) \tag{28.1}$$

$$\hat{H}_q(-y/\sigma) = (-1)^q \hat{H}_q(y/\sigma) \tag{28.2}$$

Based on Eqs. (28.1) and (28.2), the absolute values of Gaussian–Hermite polynomials are the same even the values of x and y change from negative to positive. Therefore, only one quadrant of the whole input image is required to compute the whole set of Gaussian–Hermite moments. The implementation of this symmetry property saves 75% of the required computational cost. To clarify this point and help the reader to understand it easily, the values of x and y are defined only within the range of the first quadrant (see Fig. 2) where $i = \lfloor M/2 \rfloor, \lfloor M/2 \rfloor + 1, \lfloor M/2 \rfloor + 2, \ldots, M$ and $j = \lfloor N/2 \rfloor, \lfloor N/2 \rfloor + 1, \lfloor N/2 \rfloor + 2, \ldots, N$. The image intensity function is replaced by the augmented image intensity function which is defined according to the following equation:

$$f_A(x_i, y_j) = f_1(x_i, y_j) + (-1)^p f_2(x_i, y_j) + (-1)^{p+q} f_3(x_i, y_j) + (-1)^q f_4(x_i, y_j) \tag{29}$$

where $f_1(x_i, y_j)$ represents the image intensity function of the pixel point (x_i, y_j) in the first quadrant; the other functions $f_2(x_i, y_j)$, $f_3(x_i, y_j)$ and $f_4(x_i, y_j)$ are the image intensity functions for the corresponding pixel points in the second, third and fourth quadrants respectively.

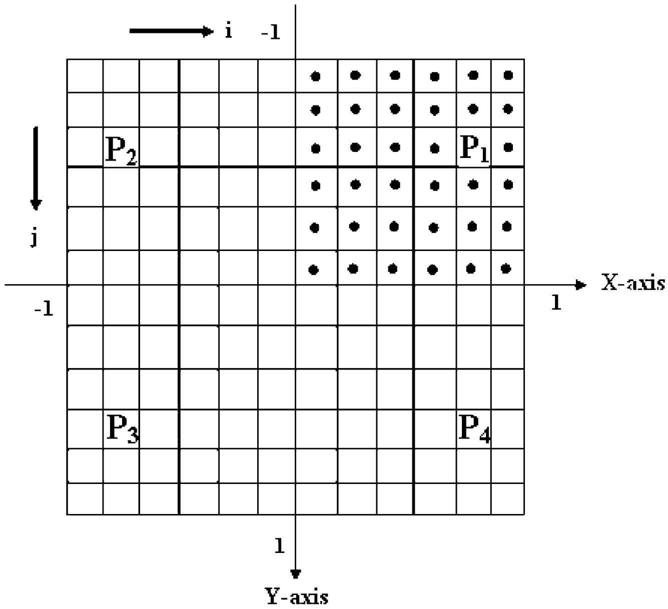

Fig. 2. The input image defined with one quadrant using the symmetry property.

3.2.2. Fast computation

The computational complexity of the accurate Gaussian–Hermite moments using Eq. (14.1) could be significantly reduced by successive computation of the 1D cascade. This methodology was proposed by Hosny in [26] and proved to be very efficient by using the numerical experiments and the theoretical complexity analysis. Based on the methodology of the 1D cascade, Eq. (14.1) will be rewritten in a separable form as follows:

$$M_{pq} = \sum_{i=\lfloor \frac{M}{2} \rfloor}^{M} IX_p(x_i) Y_{iq} \tag{30}$$

where

$$Y_{iq} = \sum_{j=\lfloor \frac{N}{2} \rfloor}^{N} IY_q(y_j) f_A(x_i, y_j) \tag{31}$$

3.2.3. Over-whole algorithm

In order to explore the proposed method and make its implementation more easily, an over-whole algorithm of the executed code for computing 2D Gaussian–Hermite moments is presented. The detailed algorithm is designed for a gray-level image of size $N \times N$ and a maximum moment order equal to *Max*. The implemented algorithm is presented in concise steps as follows:

Step 1. For $i = 1$ to $N + 1$.
Create the vectors U_i and V_j using Eqs. (18) and (19).

Step 2. For $i = 1$ to N.

Compute $x_i = \dfrac{U_{i+1} + U_i}{2}$ and $y_j = \dfrac{V_{j+1} + V_j}{2}$

Step 3. For $p = 0$ to *Max*.
Create the vector of the coefficients $C_p(\sigma)$ using the following recurrence relations:

$$C_0(\sigma) = \sqrt{\sigma\sqrt{\pi}}$$

$$C_p(\sigma) = 1/\sqrt{2p}\, C_{p-1}(\sigma), \quad \text{where } p \geqslant 1$$

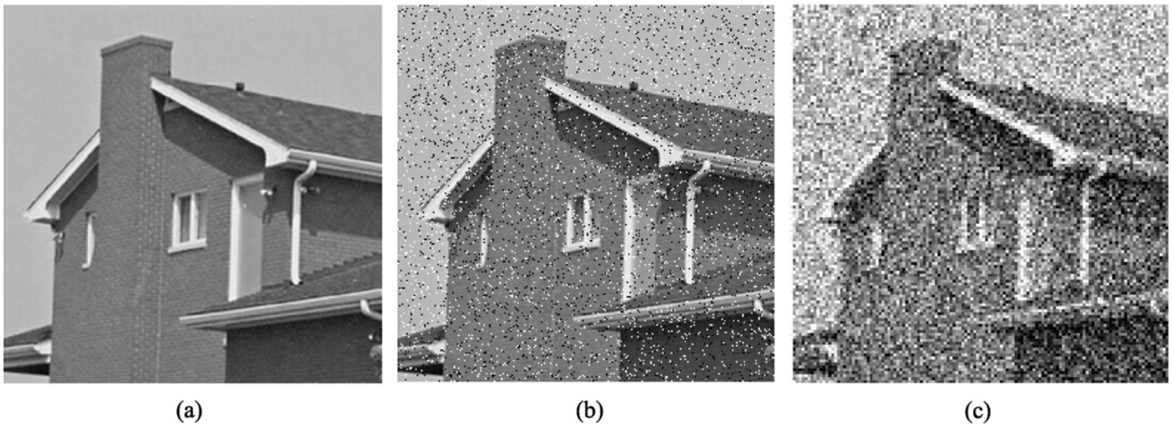

Fig. 3. House's gray-level image: (a) noise-free image, (b) noisy image: Salt & Peppers, (c) noisy image: white Gaussian.

Compute the coefficients matrix B_{pm} using the following recurrence relations:

$$B_{p0} = 2^p$$

$$B_{p,m+1} = -\frac{(p-2m-1)(p-2m)}{4(m+1)} B_{p,m}$$

Step 4. For $p = 0$ to Max & For $i = 1$ to N.
Create Hermite polynomials $H_p(x/\sigma)$ of argument (x/σ) using Eq. (3) with $H_0(x/\sigma) = 1$ and $H_2(x/\sigma) = 2x/\sigma$.

Step 5. For $p = 0$ to Max & For $i, j = 1$ to N.
Create the normalized Gaussian–Hermite polynomials $\hat{H}_p(x/\sigma)$ using Eq. (5).

Step 6. For $p = 0$ to Max.
Compute I_0 using Eqs. (24) and (25).
Compute I_1 using Eq. (26).
Compute I_T with $T \geqslant 2$ using Eq. (23).

Step 7. For $p = 0$ to Max & For $i, j = 1$ to N.
Create the kernels, $IX_p(x_i)$ and $IY_q(y_j)$, using Eqs. (20.1) and (20.2).

Step 8. For $p = 0$ to Max & For $q = 0$ to $Max - p$ with $i, j = 1$ to N.
Create the augmented image intensity function $f_A(x_i, y_j)$ using Eq. (29).

Step 9. For $p = 0$ to Max & For $q = 0$ to $Max - p$ with $i, j = \lfloor N/2 \rfloor$ to N.
Compute Y_{iq} using Eq. (31); then compute the Gaussian–Hermite moments M_{pq} using Eq. (30).

Based on the description of these steps, an elegant property could be noticed. All the steps from 1 to 7 are image-independent. Therefore, all values computed using these steps could be precomputed, saved and used whenever needed. This finding makes the proposed method very useful in real time applications and very large image databases. Only, the steps 8 and 9 are image-dependent.

4. Numerical experiments

Different numerical experiments are conducted in order to prove the validity and the efficiency of the proposed method. The detailed description of these numerical experiments will be presented in this section. The performance for the proposed method is evaluated and compared with the existing methods for computing Gaussian–Hermite moments. This section is divided into two subsections.

In the first subsection, the accuracy of the proposed method is proved by using the aspect of image reconstruction for noise-free images and images contaminated with different kinds of noise. The efficiency of the proposed method is discussed in the second subsection. CPU elapsed times are used to show the efficiency of the proposed method where these elapsed times are computed using both the proposed method and the existing conventional method [32] in the same computing environment. The CPU elapsed times are compared. Results of five numerical experiments are used to ensure the efficiency of the proposed method.

4.1. Image reconstruction

In this section, image reconstruction is used to prove the accuracy of the proposed method. Mean Square Error (MSE) between the original and the reconstructed image is widely used in the community of image processing and computer vision as a quantitative measure of the accuracy. The MSE for a digital image of size $N \times N$ is computed using the following form:

$$\mathrm{MSE} = \frac{\sum_{i=1}^{N}\sum_{j=1}^{N}(\hat{f}_{Max}(x_i, y_j) - f(x_i, y_j))^2}{N \times N} \tag{32}$$

where $f(x_i, y_j)$ is the intensity function of the original image and $\hat{f}_{Max}(x_i, y_j)$ represents the intensity function of the reconstructed image.

A numerical experiment is performed, where the 'house' gray-level image of size 128×128 as displayed in Fig. 3(a) is used with a maximum moment order ranging from 0 to 40. Fig. 4 shows the MSE of the proposed method. It is clear that, the MSE decreases as the moment order increases where the MSE approaches to zero as the moment order increases. The reconstructed image will be very close to the original one when the maximum moment order reaches a certain value.

Noise resistance of Gaussian–Hermite moments is a very desirable property. To ensure the robustness of these moments against the negative effects of different kinds of noise, two numerical experiments are conducted using noise contaminated images where two kinds of noise are used in these numerical experiments. The first kind of noise is the 'salt & pepper' noise while the second one is the strong 'white Gaussian' noise. The contaminated images with two kinds of noise are displayed in Figs. 3(b) and 3(c) respectively. The two kinds of noise are added to original noise-free images using the following Matlab8 statements:

$$A = \text{imnoise}(A, \text{'salt \& pepper'}, S) \tag{33.1}$$

$$A = \text{imnoise}(A, \text{'gaussian'}, m, v) \tag{33.2}$$

The parameters S, m and v are used to control the amount of image noise. Both noise contaminated images are reconstructed using Gaussian–Hermite moments of order ranging from 0 to 40. The plotted curves of MSE for the noise contaminated images are displayed in Fig. 4. It would be noted that, the three curves of MSE are plotted in the same figure for easier comparison.

Generally, the values of the MSE of noisy images are larger than their corresponding MSE values of noise-free image. As shown in Fig. 4, the MSE curves of the noise contaminated images approach zero by increasing the moment order. The results of these experiments ensure the robustness of Gaussian–Hermite moments against the different kinds of noise.

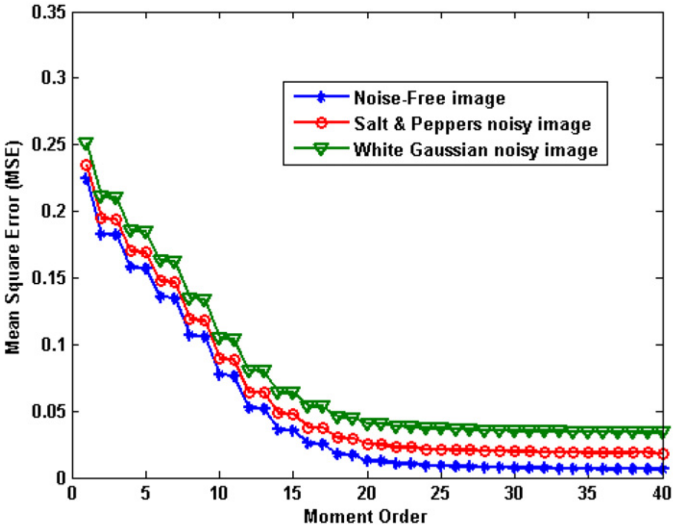

Fig. 4. MSE for the reconstructed gray-level image of House.

4.2. CPU computational time

Fast algorithms are desirable in wide rang of image processing and computer vision applications especially for real time applications and big size images. These fast algorithms are based on reduction of computational times. In order to ensure the efficiency of the proposed method, a number of numerical experiments are conducted. All of these experiments are performed using *Lenova* R4000 Laptop Machine equipped with Intel® Core™ 2 Due CPU 2.66 GHz and 3072 MB RAM and operated by 32 bit Windows 7 professional. The executed code is designed using Matlab8. The execution-time improvement ratio (ETIR) [29] is used as a criterion to compare the different computational methods. This ratio is defined as $\text{ETIR} = (1 - Time1/Time2) \times 100$, where $Time1$ and $Time2$ are the execution time of the first and the second methods. $\text{ETIR} = 0$ if both execution times are identical. The set of Gaussian–Hermite moments is computed using both Wang's [32] and the proposed methods.

In the first numerical experiment, a set of fingerprint images of unified size 128×128 as displayed in Fig. 5 is used. The set of 2D Gaussian–Hermite moments is computed using both methods. The computational processes are performed 10 times for each of the 12 fingerprint images where the average CPU elapsed times and ETIR are included in Table 1. The average elapsed times are plotted against the moment order in Fig. 6. It is clear that, the method of Wang is time-consuming and not suitable for online applications and systems where it required a higher execution time. On the other side, the proposed method is very fast method.

In the second numerical experiment, a set of standard gray-level images of size 512×512 are used. These images are displayed in Fig. 7 where the images of 'House', 'Lake', 'Baboon', and 'Mig29'

Fig. 5. Images of different fingerprints.

are displayed in the first row while the images of ‘Peppers’, ‘Pirate’, ‘Sphinx’ and the ‘Blonde women’ are displayed in the second row. The computational processes are performed and repeated 10 times for each of the 8 images where the average CPU elapsed times and the execution-time improvement ratio (ETIR) are included in Table 2. The average elapsed times are plotted against the moment order in Fig. 8. It is clear that, the proposed method tremendously reduced the execution time.

Additional numerical experiments are conducted using famous image databases. The first image database is the 53 object database

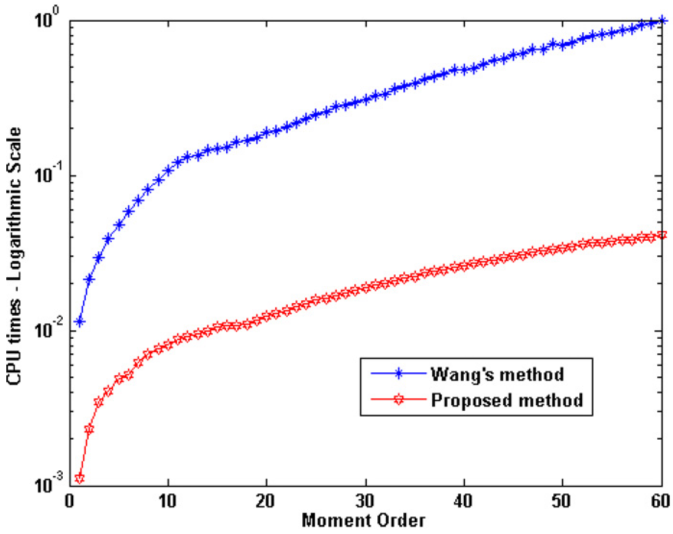

Fig. 6. Average elapsed CPU times in seconds for fingerprints.

Table 1
CPU times and reduction percentage for selected moment orders: fingerprint images of size 128×128.

Max	Wang’s method [32]	Proposed method	ETIR
1	0.0115	0.0011	90.4348%
5	0.0477	0.0042	91.1950%
10	0.1070	0.0081	92.4299%
15	0.1470	0.0104	92.9252%
20	0.1872	0.0123	93.4295%
25	0.2475	0.0156	93.6970%
30	0.3094	0.0188	93.9237%
40	0.4738	0.0263	94.4491%
50	0.6914	0.0338	95.1114%
60	0.9818	0.0411	95.8138%

Table 2
CPU times and reduction percentage for selected moment orders: gray-level images of size 512×512.

Max	Wang’s method [32]	Proposed method	ETIR
1	0.0589	0.0064	89.1341%
5	0.2166	0.0139	93.5826%
10	0.5871	0.0231	96.0654%
15	1.1194	0.0381	96.5964%
20	1.8823	0.0484	97.4287%
25	2.6949	0.0620	97.6994%
30	3.7206	0.0724	98.0541%
40	6.3408	0.0965	98.4781%
50	9.6211	0.1184	98.7694%
60	13.5195	0.1444	98.9316%

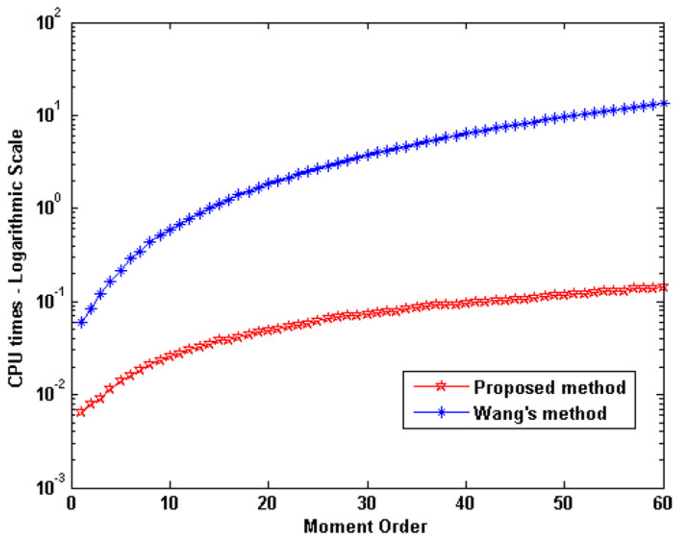

Fig. 8. Average elapsed CPU times in seconds for gray-level images.

Fig. 7. Gray-level images: House, Lake, Baboon, Mig29, Peppers, Pirate, Sphinx and Blonde women.

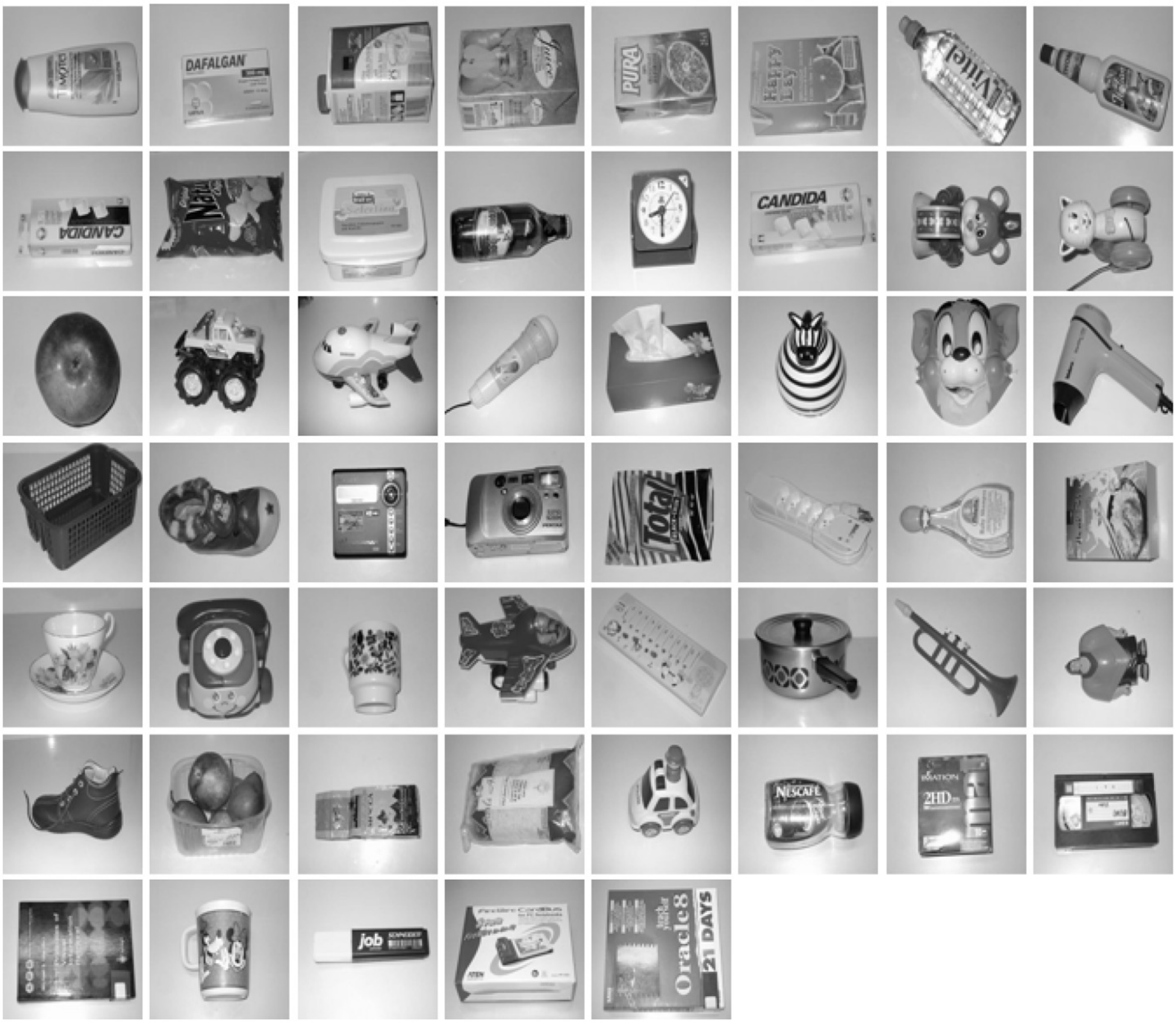

Fig. 9. Gray-level images of the 53 objects.

[34] which contains 265 images with a unified image size equal to 320×240, where each object is represented by 5 images. A gray-level image of the collection of the objects is displayed in Fig. 9. The second image database is ORL-faces database [35]. This database contains ten different images for the face of each person, where the total number of images is equal to 400. All images of this database have the size 92×112. Fig. 10 displays a collection of the 40 faces. The third image database is the Columbia Object Image Library (COIL-20) database [36]. The total number of images is 1440 distributed as 72 images for each object. All images of this database have the size 416×448. Fig. 11 displays a collection of the 20 objects.

The Gaussian–Hermite moments of these databases are computed using both Wang's and the proposed method. Similar to the previous numerical experiment, the computational process is repeated 10 times and the average execution times of both methods are computed. The obtained results are represented and displayed in Fig. 12. The results of these additional numerical experiments are consistent with the results of the previously conducted numerical experiments. The execution times of the Wang's method are much higher than the corresponding execution times of the proposed method. Based on the results of the conducted experiments, the proposed method is a real time method while the method of Wang is impractical for large images.

5. Conclusion

This paper proposes a method for fast and efficient computation of highly accurate Gaussian–Hermite moments for binary and gray-level images. The extremely fast computation and low complexity requirements of the proposed Gaussian–Hermite moments are suitable for handling the large databases of digital images and online computer vision applications.

Fig. 10. Collection of the 40 faces (ORL-faces).

Fig. 11. Collection of the COIL-20 objects.

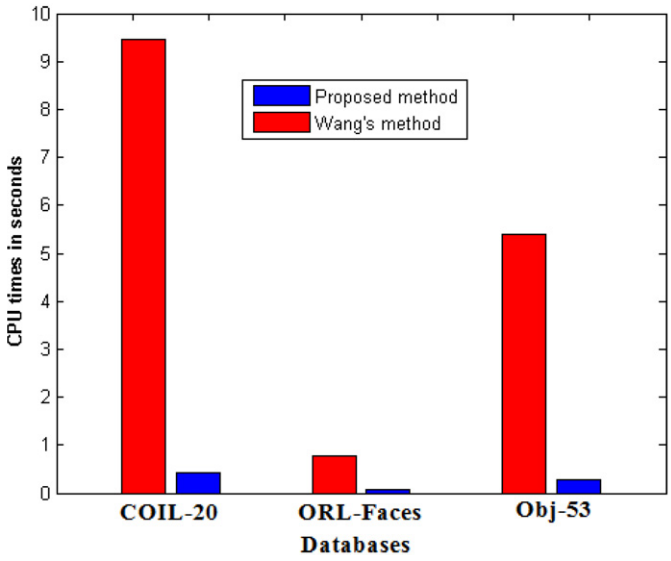

Fig. 12. Average elapsed CPU times in seconds for the three databases.

References

[1] W.Y. Kim, H.J. Kim, Eye detection in facial images using Zernike moments with SVM, ETRI J. 30 (2008) 335–337.
[2] V.S. Bharathi, L. Ganesan, Orthogonal moments based texture analysis of CT liver images, Pattern Recogn. Lett. 29 (2008) 1868–1872.
[3] T.J. Bin, A. Lei, C.J. Wen, K. Wen-jing, L. Dan-dan, Subpixel edge location based on orthogonal Fourier–Mellin moments, Image Vision Comput. 26 (2008) 563–569.
[4] A. Broumandnia, J. Shanbehzadeh, Fast Zernike wavelet moments for Farsi character recognition, Image Vision Comput. 25 (2007) 717–726.
[5] K.M. Hosny, Robust template matching using orthogonal Legendre moment invariants, J. Comput. Sci. 6 (10) (2010) 1083–1087.
[6] Ismail A. Ismail, Mohamed A. Shouman, Khalid M. Hosny, Hayam M. Abdel Salam, Invariant image watermarking using accurate Zernike moments, J. Comput. Sci. 6 (1) (2010) 52–59.
[7] M.K. Hu, Visual pattern recognition by moment invariants, IRE Trans. Inform. Theory 8 (1) (1962) 179–187.
[8] M.R. Teague, Image analysis via the general theory of moments, J. Opt. Soc. Am. 70 (8) (1980) 920–930.
[9] C.H. Teh, R.T. Chin, On image analysis by the method of moments, IEEE Trans. Pattern Anal. Mach. Intell. 10 (4) (1988) 496–513.
[10] J. Shen, Orthogonal Gaussian–Hermite moments for image characterization, in: Proc. SPIE Intelligent Robots Computer Vision XVI: Algorithms, Techniques, Active Vision, and Materials Handling, vol. 3280, Pittsburgh, PA, USA, October 1997, pp. 224–233.
[11] J. Shen, W. Shen, D.-F. Shen, On geometric and orthogonal moments, Int. J. Pattern Recogn. Artif. Intell. 14 (7) (2000) 875–894.
[12] Youfu Wu, June Shen, Properties of orthogonal Gaussian–Hermite moments and their application, EURASIP J. Appl. Signal Process. 4 (2005) 1–12.
[13] J. Shen, W. Shen, D.-F. Shen, Y. Wu, Orthogonal moments and their application to motion detection in image sequences, Int. J. Inform. Acquis. 1 (1) (2004) 77–87.
[14] Y. Wu, J. Shen, Moving object detection using orthogonal Gaussian–Hermite moments, in: Proceedings of SPIE in Visual Communications and Image, vol. 5308, San Jose, CA, USA, January 2004, pp. 841–849.
[15] Youfu Wu, Jun Shen, Mo Dai, Traffic object detections and its action analysis, Pattern Recogn. Lett. 26 (13) (2005) 1963–1984.
[16] L. Wang, H. Sau, M. Dai, Fingerprint image segmentation based on Gaussian–Hermite moments, in: Lecture Notes in Computer Science, vol. 3584, 2005, pp. 446–454.
[17] Lin Wang, Mo Dai, An effective method for extracting singular points in fingerprint images, Int. J. Electron. Commun. (AEU) 60 (9) (2006) 671–676.
[18] Lin Wang, Mo Dai, Application of a new type of singular points in fingerprint classification, Pattern Recogn. Lett. 28 (2007) 1640–1650.
[19] X. Zheng, M. Dai, M. Zhou, Gauss–Hermite moments application in medical image segmentation, in: 5th International Conference on Visual Information Engineering (VIE 2008), 2008, pp. 1–5.
[20] W. Shen, Y. Xiao, M. Daoudi, Stereo matching based on orthogonal Gaussian–Hermite moments, in: 6th International Symposium on Multispectral Image Processing and Pattern Recognition, Proc. SPIE, vol. 7496, 2009.
[21] S.M. Mahbubur Rahman, M. Omair Ahmad, M.N.S. Swamy, Bayesian wavelet-based image denoising using the Gauss–Hermite expansion, IEEE Trans. Image Process. 17 (10) (2008) 1755–1771.
[22] L. Ma, T.N. Tan, Y.H. Wang, D.X. Zhang, Local intensity variation analysis for iris recognition, Pattern Recogn. 37 (6) (2004) 1287–1298.
[23] X. Ma, R. Pan, L. Wang, License plate character recognition based on Gaussian–Hermite moments, in: Proceedings of the Second International Workshop on Education Technology and Computer Science, vol. 3, 2010, pp. 11–14.
[24] Bo Yang, Gengxiang Li, Huilong Zhang, Mo Dai, Rotation and translation invariants of Gaussian–Hermite moments, Pattern Recogn. Lett. 32 (2011) 1283–1298.
[25] S.X. Liao, M. Pawlak, On image analysis by moments, IEEE Trans. Pattern Anal. Mach. Intell. 18 (3) (1996) 254–266.
[26] K.M. Hosny, Exact and fast computation of geometric moments for gray level images, Appl. Math. Comput. 189 (2007) 1214–1222.
[27] K.M. Hosny, Exact Legendre moment computation for gray level images, Pattern Recogn. 40 (12) (2007) 3597–3605.
[28] K.M. Hosny, Fast and accurate method for radial's moments computation, Pattern Recogn. Lett. 31 (2) (2010) 143–150.
[29] K.M. Hosny, Fast computation of accurate Zernike moments, J. Real-Time Image Process. 3 (1–2) (2008) 97–107.
[30] K.M. Hosny, A systematic method for efficient computation of full and subsets Zernike moments, Inform. Sci. 180 (11) (2010) 2299–2313.
[31] M. Abramowitz, I.A. Stegun, Handbook of Mathematical Functions, Dover Publications, Inc., New York, 1965.
[32] Lin Wang, Youfu Wu, Mo Dai, Some aspects of Gaussian–Hermite moments in image analysis, in: IEEE Third International Conference on Natural Computation (ICNC), vol. 2, 2007, pp. 450–454.
[33] K.M. Hosny, Image representation using accurate orthogonal Gegenbauer moments, Pattern Recogn. Lett. 32 (6) (2011) 795–804.
[34] http://www.vision.ee.ethz.ch/datasets/index.en.html.
[35] http://www.cl.cam.ac.uk/research/dtg/attarchive/facedatabase.html.
[36] http://www.cs.columbia.edu/CAVE/software/softlib/coil-20.php.

Khalid M. Hosny received the B.Sc., M.Sc. and Ph.D. from Zagazig University, Zagazig, Egypt in 1988, 1994, and 2000 respectively. From 1997 to 1999 he was a visiting scholar, University of Michigan, Ann Arbor and University of Cincinnati, Cincinnati, USA. He joined the faculty of Computers and Informatics at Zagazig University, where he held the position of associate professor. He is a senior member of ACM and IEEE and the IEEE Computer Society. His research interests include image processing, pattern recognition and computer vision. Dr. Hosny published more than thirty papers in international journals. He is a member of the editorial board of Journal of Pattern Recognition Research and the International Journal of Image Processing. Also, he works as a reviewer for the Image and Vision Computing Journal, Information Science Journal, Signal Processing Journal, Pattern Recognition Letters, Journal of Computational Statistics and Data Analysis, Journal of Systems and Software and Applied Mathematics and Computation, Journal of Multimedia Tools and Applications, Journal of Real-Time Image Processing, Journal of Mathematical Imaging and Vision, IET Image Processing Journal, Optics Express, Information Technology Journal, Applied Science Journal and Imaging Science Journal, International Journal of Signal and Imaging Systems Engineering.