Hindawi Publishing Corporation
Mathematical Problems in Engineering
Volume 2012, Article ID 353406, 17 pages
doi:10.1155/2012/353406

Research Article

An Algorithm for Fast Computation of 3D Zernike Moments for Volumetric Images

Khalid M. Hosny[1,2] and Mohamed A. Hafez[3]

[1] *Department of Computer Science, Community College, Najran University, P.O. BOX 1988, Najran, Saudi Arabia*
[2] *Department of Information Technology, Faculty of Computers and Informatics, Zagazig University, Zagazig 44519, Egypt*
[3] *Department of Mathematics, College of Science and Arts, Najran University, P.O. BOX 1988, Najran, Saudi Arabia*

Correspondence should be addressed to Khalid M. Hosny, k_hosny@yahoo.com

Received 9 May 2012; Revised 6 August 2012; Accepted 29 August 2012

Academic Editor: Wanquan Liu

An algorithm was proposed for very fast and low-complexity computation of three-dimensional Zernike moments. The 3D Zernike moments were expressed in terms of exact 3D geometric moments where the later are computed exactly through the mathematical integration of the monomial terms over the digital image/object voxels. A new symmetry-based method was proposed to compute 3D Zernike moments with 87% reduction in the computational complexity. A fast 1D cascade algorithm was also employed to add more complexity reduction. The comparison with existing methods was performed, where the numerical experiments and the complexity analysis ensured the efficiency of the proposed method especially with image and objects of large sizes.

1. Introduction

Moments of images are generally defined as projections of the image function onto a set of basis functions. In his pioneer work, Hu [1] popularized the usage of image moments in 2D pattern recognition. The set of Hu's moments gain the interest of scientists and widely applied during the last five decades. Teague [2] suggested the usage of orthogonal basis functions such as Legendre and Zernike to construct moments. Orthogonal moments are used to represent images with a minimum amount of information redundancy. Teh and Chin [3] evaluated different orthogonal and nonorthogonal moments. They found 2D Zernike moments superior over others moments in terms of noise sensitivity and discrimination power. 2D Zernike moments are used in a wide range of applications such as pattern

recognition applications [4–6], content-based image retrieval [7–9], image watermarking [10–12], biometrics [13, 14], analysis and recognition of medical images [15–17], and edge detection [18].

Reconstruction, recognition, and discrimination of 3D objects gained more research interest during the last decade. 3D moment invariants are used as shape descriptors [19]. The superiority of orthogonal 2D Zernike moments over the nonorthogonal moments motivate Canterakis [20] to generalize the classical 2D Zernike polynomials to 3D. Canterakis paid his attention to theoretical aspects of deriving 3D Zernike polynomials and moments. Novotni and Klein [21] addressed 3D Zernike descriptors for content-based shape retrieval where the information content of the recovered 3D shape has no redundancy because of the orthonormality. In addition to this property, a group of 3D Zernike moments are rotation-invariant and the shape reconstruction from 3D Zernike is a very simple process. Also, 3D Zernike moments have the advantage of capturing global information about the 3D shape without requiring closed boundaries as in boundary-based methods.

Due to the attractive properties, applications of 3D Zernike descriptor gain more interest. Millán et al. [22] used 3D Zernike moment invariants in morphological characterization of intracranial aneurysms. Qiuting and Bing [23] used the 3D Zernike moments to derive a 3D terrain matching algorithm. Since, shape plays a crucial role in molecular recognition and function. The development of shape analysis techniques is important for understanding protein structure-function relationships. Recently, 3D Zernike moments gain more interest from peoples working in the area of bioinformatics and molecular biology. Sael et al. [24, 25] applied 3D Zernike moments for protein tertiary structure retrieval and comparison of properties on protein surface. 3D Zernike moments are promising descriptors in the field of biological imaging. X-ray diffraction and electronic microscopic imaging are examples where these 3D descriptors could be used to build up 3D views of biological entities such as proteins, nucleic acids, cells, tumors, tissues and whole organs or organisms.

Unfortunately, high computational demands of 3D Zernike moments hindered the wide applications of these applications. The need for computational approaches to efficiently store, display, and compare this data is the motivation of the present work. In the literature, there are many methods and approaches for fast computation of 2D Zrnike moments. The extendibility of these methods to compute 3D Zernike moments will be discussed.

In the first approach, Papakostas et al. [26] proposed a modified direct method for the computation of the Zernike moments. They computed factorial terms by using Sirling's formula. This method cannot be extended to compute 3D Zernike moments. Using the recurrence relations is another approach that should be consider. Al-Rawi [27] proposed an algorithm for fast computation of 2D Zernike moments. Unfortunately, this approach cannot be extended to compute 3D Zernike moments.

Exact computation of 2D Zernike moments via exact 2D geometric moments is the third approach. Wee and Paramesran [28] proposed a novel method in which they expressed 2D Zernike moments as expansion of exact 2D geometric moments of the same order. In fact, the method of Chong is highly accurate and very time consuming. Hosny [29] modified the method of Chong to be a very fast method. Recently, Hosny [30] proposed a novel method for fast and accurate computation of the full and subsets 2D Zernike moments. Fortunately, this approach is extendable to compute 3D Zernike moments.

In this work, a fast, low-complexity method is proposed for efficient computation of 3D Zernike moments. The entire set of 3D Zernike moments and the selected set of rotationally invariant 3D Zernike moments are computed as a combination of exact 3D geometric moments. A 3D symmetry-based method is applied where 87% of the computational

complexity is reduced. A fast algorithm is applied to accelerate the computational process. The heavy computational expensive binomial coefficients terms are avoided by using a very simple computational method. The proposed method significantly reduces the whole computational complexity and seems to be more suitable for large objects and databases.

2. Three-Dimensional Zernike Moments

Three-dimensional Zernike polynomials, $Z_{n,\ell,m}$, are orthogonal polynomials defined on a unit ball as follows:

$$Z_{n,\ell,m}(\Re) = R_{n,\ell}(r)Y_{\ell,m}(\theta,\phi), \tag{2.1}$$

where $n \in [0, \text{Max}]$, $\ell \in [0, n]$, and $m \in [-\ell, \ell]$. The value $(n - \ell)$ must be even nonnegative integer number. Max is the maximum order considered in the computational process. $R_{n,\ell}(r)$ and $Y_{\ell,m}(\theta,\phi)$ are real-valued radial functions and spherical Harmonics, respectively. Any function, $f(\Re)$, defined within the unit ball could be expanded by using the 3D Zernike polynomials as follows:

$$f(\Re) = \sum_{n=0}^{\infty} \sum_{\ell=0}^{n} \sum_{m=-\ell}^{\ell} \Omega_{n,\ell,m} Z_{n,\ell,m}(\Re). \tag{2.2}$$

The expansion coefficients, $\Omega_{n,\ell,m}$, represent the 3D Zernike moments and determined by using the complex conjugate of 3D Zernike polynomials as follows:

$$\Omega_{n,\ell,m} = \int_0^1 \int_0^{2\pi} \int_0^{\pi} \overline{Z_{n,\ell,m}(\Re)} f(\Re) r^2 \sin\theta \, dr \, d\theta \, d\phi. \tag{2.3}$$

According to the extreme complexity of computation in the 3D spherical coordinates, Canterakis [20] formulated the 3D Zernike polynomials in the Cartesian coordinates where the conversion between spherical and Cartesian coordinates, $[X]^T = [\Re]^T$, is defined in an explicit form as follows:

$$\begin{bmatrix} x \\ y \\ z \end{bmatrix} = \begin{bmatrix} r\sin\theta\sin\phi \\ r\sin\theta\cos\phi \\ r\cos\phi \end{bmatrix}. \tag{2.4}$$

The 3D Zernike polynomials in the Cartesian coordinates are defined as follows:

$$Z_{n,\ell,m}(X) = \sum_{v=0}^{k} Q_{k,\ell,v} |X|^{2v} e_{\ell,m}(X), \tag{2.5}$$

where k is an integer equal to $(n-\ell)/2$ and $0 \le v \le k$. The coefficients $Q_{k,\ell,v}$ are defined as

$$Q_{k,\ell,v} = \frac{(-1)^k}{2^{2k}}\sqrt{\frac{2\ell+4k+3}{3}}\binom{2k}{k}(-1)^v \frac{\binom{k}{v}\binom{2(k+\ell+v)+1}{2k}}{\binom{k+\ell+v}{k}}. \tag{2.6}$$

The harmonic polynomials, $e_{\ell,m}(\mathrm{X})$, are defined as

$$e_{\ell,m}(\mathrm{X}) = C_{\ell,m} r^{\ell}\left(\frac{\widehat{i}x-y}{2}\right)^m z^{\ell-m}\sum_{\mu=0}^{\lfloor(\ell-m)/2\rfloor}\binom{\ell}{\mu}\binom{\ell-\mu}{m+\mu}\left(-\frac{x^2+y^2}{4z^2}\right)^{\mu}, \tag{2.7}$$

where $\widehat{i} = \sqrt{-1}$, $z = x + \widehat{i}y$ is the complex variable and $C_{\ell,m}$ are the normalization factors and defined as

$$C_{\ell,m} = \frac{\sqrt{(2\ell+1)(\ell+m)!(\ell-m)!}}{\ell!}. \tag{2.8}$$

Based on (2.8), the normalization factors $C_{\ell,m}$ of positive and negative values of m are identical, $C_{\ell,-m} = C_{\ell,m}$ while the harmonic polynomials with negative values of m are defined as

$$e_{\ell,-m}(\mathrm{X}) = (-1)^m \overline{e_{\ell,m}(\mathrm{X})}. \tag{2.9}$$

The orthogonality relation of 3D Zernike polynomials is defined as follows:

$$\frac{3}{4\pi}\int_{|\mathrm{X}|\le 1} Z_{n,\ell,m}(\mathrm{X})\overline{Z_{n',\ell',m'}(\mathrm{X})}d\mathrm{X} = \delta_{n,n'}\delta_{\ell,\ell'}\delta_{m,m'}. \tag{2.10}$$

The 3D Zernike moments and the image/object reconstruction in spherical coordinates as defined by (2.3) and (2.2) are converted into the Cartesian coordinates and rewritten as follows:

$$\begin{aligned}\Omega_{n,\ell,m} &= \frac{3}{4\pi}\int_{|\mathrm{X}|\le 1} f(\mathrm{X})\overline{Z_{n,\ell,m}(\mathrm{X})}d\mathrm{X}\\ f(\mathrm{X}) &= \sum_{n=0}^{\infty}\sum_{\ell=0}^{n}\sum_{m=-\ell}^{\ell}\Omega_{n,\ell,m}Z_{n,\ell,m}(\mathrm{X}),\end{aligned} \tag{2.11}$$

where the 3D Zernike moments with negative values of m could be computed directly from their corresponding ones with positive values of m as follows:

$$\Omega_{n,\ell,-m}(\mathrm{X}) = (-1)^m\overline{\Omega_{n,\ell,m}(\mathrm{X})}. \tag{2.12}$$

Canterkis [20] compactly formulate the 3D Zernike polynomials of order n as a linear combination of monomials of order up to n as follows:

$$Z_{n,\ell,m}(\mathrm{X}) = \sum_{r+s+t\leq n} H^{rst}_{n,\ell,m} x^r y^s x^t. \tag{2.13}$$

Consequently, the 3D Zernike moments $\Omega_{n,\ell,m}$ of order n could be expressed as a linear combination of the 3D geometric moments, $G_{r,s,t}$, by using the following relation:

$$\Omega_{n,\ell,m} = \frac{3}{4\pi} \sum_{r+s+t\leq n} \overline{H^{rst}_{n,\ell,m}} G_{r,s,t}, \tag{2.14}$$

where the complex coefficients, $H^{rst}_{n,\ell,m}$, and the 3D geometric moments are defined as

$$\begin{aligned} H^{rst}_{n,\ell,m} = {} & C_{\ell,m} 2^{-m} \cdot \sum_{v=0}^{k} Q_{k,\ell,v} \cdot \sum_{\alpha=0}^{v} \binom{v}{\alpha} \cdot \sum_{\beta=0}^{v-\alpha} \binom{v-\alpha}{\beta} \cdot \sum_{u=0}^{m} (-1)^{m-u} \binom{m}{u} \left(\widehat{i}\right)^u \\ & \cdot \sum_{\mu=0}^{\lfloor (\ell-m)/2 \rfloor} (-1)^{\mu} 2^{-2\mu} \binom{\ell}{\mu} \binom{\ell-\mu}{m+\mu} \cdot \sum_{\eta=0}^{\mu} \binom{\mu}{\eta}, \end{aligned} \tag{2.15}$$

$$G_{r,s,t} = \int_{|\mathrm{X}|\leq 1} f(\mathrm{X}) x^r y^s x^t d\mathrm{X}, \tag{2.16}$$

with $r = 2(\eta + \alpha) + u$, $s = 2(\mu - \eta + \beta) + m - u$, and $t = 2(v - \alpha - \beta - \mu) + \ell - m$.

The rotation invariance of 3D Zernike moments could be achieved where the moments are collected into $(2\ell + 1)$ vectors $\Omega_{n,\ell} = (\Omega_{n,\ell,\ell}, \Omega_{n,\ell,\ell-1}, \Omega_{n,\ell,\ell-2}, \ldots, \Omega_{n,\ell,-\ell})^T$. The rotational invariant 3D Zernike descriptors, $F_{n,\ell}$, are defined as the norms of these vectors. To avoid any misunderstanding, an example is illustrated. For maximum moment order, Max = 10, the total number of 3D Zernike moments is equal to 286. The number of independent 3D Zernike moments is 161, while the number of 3D Zernike descriptors for this moment order is 36. The number of 3D Zernike descriptors could be easily determined using the following form:

$$\text{Total} = \begin{cases} \left(\dfrac{\text{Max}+2}{2}\right)^2, & \text{Max is even} \\ \dfrac{(\text{Max}+1)(\text{Max}+3)}{4}, & \text{Max is odd.} \end{cases} \tag{2.17}$$

The independent 3D Zernike moments are used in 3D image reconstruction, while the 3D Zernike descriptors are used in comparing similar structure by simply comparing the vectors of these descriptors. Theoretically, if a 3D object is rotated with any angle, the vectors of 3D Zernike descriptors must be the same for both original and rotated objects.

3. The Proposed Algorithm

3D Zernike moments are defined as the projection of the digital image/object onto the 3D Zernike polynomials. These polynomials are defined in spherical coordinates within a unit ball. By converting to Cartesian coordinates, the 3D image/object is defined within a cube; this cube is completely surrounded by this unit ball where its centre coincides with the centre of the unit ball. The coordinate axes x, y, and z divide the mentioned cube into eight equal small cubes.

The 3D digital image/object of size $N \times N \times N$ is a multidimensional array of voxels; centers of these voxels are the points (x_i, y_j, z_k) where the intensity function is defined only for this discrete set of points $(x_i, y_j, z_k) \in \lfloor -1/\sqrt{3}, 1/\sqrt{3} \rfloor \times \lfloor -1/\sqrt{3}, 1/\sqrt{3} \rfloor \times \lfloor -1/\sqrt{3}, 1/\sqrt{3} \rfloor$ with:

$$x_i = \frac{2i - N - 1}{N\sqrt{3}}, \qquad y_j = \frac{2j - N - 1}{N\sqrt{3}}, \qquad z_k = \frac{2k - N - 1}{N\sqrt{3}}, \tag{3.1}$$

where the sampling intervals in the x-, y-, and z-directions are $\Delta x_i = x_{i+1} - x_i$, $\Delta y_j = y_{j+1} - y_j$, and $\Delta z_k = z_{k+1} - z_k$ with $i, j, k = 1, 2, \ldots, N$.

Computation of 3D Zernike moments in Cartesian coordinated is completely dependent on two computational modules. The first one consists of the computational processes required to compute the complex coefficients $H^{rst}_{n,\ell,m}$. The second is the process of computing the 3D geometric moments. In order to reduce the overall computational complexity, both modules must be designed and executed by using efficient methodology.

The extremely time-consuming computational process of the first module could be significantly reduced through the recurrence relations and avoiding the repeated evaluation of factorial terms. The computational complexity of the second module is significantly reduced through the implementation of a symmetry property and the successive computation of 1D cascade for each moment order. Through the next subsection, a detailed description of the proposed efficient method is presented.

3.1. Computational Aspects

Fast algorithms are generally desired in the computational processes of 3D moments. In this subsection, the attention is paid to reduce the computational complexity of the first module. Analysis of (2.15) shows that the computation of the complex coefficients $H^{rst}_{n,\ell,m}$ for each moment order required the computation of the normalization factors $C_{\ell,m}$, the coefficients $Q_{k,\ell,v}$, and a huge number of factorial terms. Computation of each of these coefficients is a source of excessive computational complexity. These time-consuming processes are repeated with each moment order. In order to reduce the overall computational complexity, an efficient method must be applied to overcome these aforementioned sources of computational complexity.

A recurrence relation is derived and employed to compute the normalization factors $C_{\ell,m}$. The derived relation completely ignored the heavy computational costs of factorial terms. It is clear that these normalization factors are image-independent; so, their values

could be precalculated, stored, and recalled whenever they are needed. This is another attractive point. According to (2.8), the derived recurrence relations are

$$C_{0,0} = 1, \tag{3.2}$$

$$C_{\ell,0} = \sqrt{2\ell + 1}, \tag{3.3}$$

$$C_{\ell,m} = \sqrt{\frac{\ell + m}{\ell - m + 1}} C_{\ell,m-1}, \tag{3.4}$$

where $\ell = 1, 2, 3, \ldots \text{Max}$ and $m = 1, 2, 3, \ldots \ell$.

The second source of excessive computational complexity is the coefficients $Q_{k,\ell,v}$ defined by (2.6). To overcome this problem, (2.6) will be rewritten as follows:

$$Q_{k,\ell,v} = \frac{(-1)^{k+v}}{2^{2k}} \sqrt{\frac{2\ell + 4k + 3}{3}} T_{k,\ell,v}, \tag{3.5}$$

with

$$T_{k,\ell,v} = \frac{\binom{2k}{k}\binom{k}{v}\binom{2(k+\ell+v)+1}{2k}}{\binom{k+\ell+v}{k}}. \tag{3.6}$$

The time-consuming direct computations of combinational terms are avoided by using the recurrence relations [29] where an image-independent matrix could be precomputed, stored, and recalled whenever it is needed. The combinational terms of (3.6) could be easily computed using the stored values of this matrix. For moment order $n \in [0, \text{Max}]$, $\ell \in [0, n]$, k is an integer equal to $k = (n - \ell)/2$ and $0 \leq v \leq k$, the numerical values of the coefficients $T_{k,\ell,v}$ will be precalculated, stored, and recalled any time needed. Consequently, the coefficients $Q_{k,\ell,v}$ could be efficiently computed without any combination or factorial terms. So, we could overcome the second source of excessive computational complexity. For the special case, $\ell = n$, the coefficients $Q_{k,\ell,v}$ are easily computed using the following equation:

$$Q_{0,\ell,0} = \sqrt{\frac{2\ell + 3}{3}}. \tag{3.7}$$

To see the importance of (3.7), the 3D image reconstruction using 3D Zernike moments of maximum order 10 required the computation of 161 independent moments. Equation (3.7) computes 66 of these moments where the highest number of moments is found when the order n equals to ℓ.

The third source of computational complexity is the direct computation of the complex coefficients $H^{rst}_{n,\ell,m}$ using (2.15). Factorial terms represent the main source of the complexity in (2.15) where ignoring these factorial terms results in a great simplicity. Replacing factorial terms in (2.15) by their numerical values will achieve this goal.

Table 1: 3D symmetry points and their coordinates.

Points	Coordinates
P_1	$(x_i\ , y_j,\ z_k)$
P_2	$(x_{N-i+1}\ , y_j,\ z_k)$
P_3	$(x_{N-i+1}\ , y_{N-j+1},\ z_k)$
P_4	$(x_i\ , y_{N-j+1},\ z_k)$
P_5	$(x_i\ , y_j,\ z_{N-k+1})$
P_6	$(x_{N-i+1}\ , y_j,\ z_{N-k+1})$
P_7	$(x_{N-i+1}\ , y_{N-j+1},\ z_{N-k+1})$
P_8	$(x_i\ , y_{N-j+1},\ z_{N-k+1})$

3.2. Symmetry Property

According to the mapping of the 3D object inside the unit ball, the radial distance from any point $P_1(x_i, y_j, z_k)$ to the coordinate origin is $\sqrt{{x_i}^2 + {y_j}^2 + {z_k}^2}$. Based on the definition of radial distance, there are eight points in the different eight small cubes having the same radial distance to the coordinate origin. These points and their coordinates are shown in Table 1.

Since the points $\{P_d,\ d = 1,2,3,\ldots,8\}$ have the same radial distance, then the numerical values of $x^r\ y^s\ z^t$ will be dependent on whatever r, s, and t are even or odd. For quick proof of this assumption, we consider the following illustrative example. Let the first point P_1 has the coordinates $P_1(x_4, y_3, z_2) \equiv P_1(-1/8\sqrt{3}, -3/8\sqrt{3}, -5/8\sqrt{3})$. Based on coordinate relations defined in Table 1, the other seven points could be written as $P_2(1/8\sqrt{3}, -3/8\sqrt{3}, -5/8\sqrt{3})$, $P_3(1/8\sqrt{3}, 3/8\sqrt{3}, -5/8\sqrt{3})$, $P_4(-1/8\sqrt{3}, 3/8\sqrt{3}, -5/8\sqrt{3})$, $P_5(-1/8\sqrt{3}, -3/8\sqrt{3}, 5/8\sqrt{3})$, $P_6(1/8\sqrt{3}, -3/8\sqrt{3}, 5/8\sqrt{3})$, $P_7(1/8\sqrt{3}, 3/8\sqrt{3}, 5/8\sqrt{3})$, and $P_8(-1/8\sqrt{3}, 3/8\sqrt{3}, 5/8\sqrt{3})$. Numerical values of $x^r\ y^s\ z^t$ for the points $\{P_d,\ d = 1,2,3,\ldots,8\}$ with different possibilities of exponent indices, r, s, and t are listed in Table 2.

Based on this symmetry property and the numerical results in Table 2, we can define the different eight cases for what is called augmented intensity function as follows.

Case 1. $r = E$, $s = E$, and $t = E$;

$$\begin{aligned} f_A(x_i, y_j, z_k) = {} & f_1(x_i, y_j, z_k) + f_2(x_i, y_j, z_k) + f_3(x_i, y_j, z_k) + f_4(x_i, y_j, z_k) \\ & + f_5(x_i, y_j, z_k) + f_6(x_i, y_j, z_k) + f_7(x_i, y_j, z_k) + f_8(x_i, y_j, z_k), \end{aligned} \tag{3.8a}$$

Case 2. $r = E$, $s = E$, and $t = O$;

$$\begin{aligned} f_A(x_i, y_j, z_k) = {} & -f_1(x_i, y_j, z_k) - f_2(x_i, y_j, z_k) - f_3(x_i, y_j, z_k) - f_4(x_i, y_j, z_k) \\ & + f_5(x_i, y_j, z_k) + f_6(x_i, y_j, z_k) + f_7(x_i, y_j, z_k) + f_8(x_i, y_j, z_k), \end{aligned} \tag{3.8b}$$

Case 3. $r = E$, $s = O$, and $t = E$;

$$\begin{aligned} f_A(x_i, y_j, z_k) = {} & -f_1(x_i, y_j, z_k) - f_2(x_i, y_j, z_k) + f_3(x_i, y_j, z_k) + f_4(x_i, y_j, z_k) \\ & - f_5(x_i, y_j, z_k) - f_6(x_i, y_j, z_k) + f_7(x_i, y_j, z_k) + f_8(x_i, y_j, z_k), \end{aligned} \tag{3.8c}$$

Case 4. $r = E$, $s = O$, and $t = O$;

$$\begin{aligned} f_A(x_i, y_j, z_k) = {} & f_1(x_i, y_j, z_k) + f_2(x_i, y_j, z_k) - f_3(x_i, y_j, z_k) - f_4(x_i, y_j, z_k) \\ & - f_5(x_i, y_j, z_k) - f_6(x_i, y_j, z_k) + f_7(x_i, y_j, z_k) + f_8(x_i, y_j, z_k), \end{aligned} \tag{3.8d}$$

Case 5. $r = O$, $s = E$, and $t = E$;

$$\begin{aligned} f_A(x_i, y_j, z_k) = {} & -f_1(x_i, y_j, z_k) + f_2(x_i, y_j, z_k) + f_3(x_i, y_j, z_k) - f_4(x_i, y_j, z_k) \\ & - f_5(x_i, y_j, z_k) + f_6(x_i, y_j, z_k) + f_7(x_i, y_j, z_k) - f_8(x_i, y_j, z_k), \end{aligned} \tag{3.8e}$$

Case 6. $r = O$, $s = E$, and $t = O$;

$$\begin{aligned} f_A(x_i, y_j, z_k) = {} & f_1(x_i, y_j, z_k) - f_2(x_i, y_j, z_k) - f_3(x_i, y_j, z_k) + f_4(x_i, y_j, z_k) \\ & - f_5(x_i, y_j, z_k) + f_6(x_i, y_j, z_k) + f_7(x_i, y_j, z_k) - f_8(x_i, y_j, z_k), \end{aligned} \tag{3.8f}$$

Case 7. $r = O$, $s = O$, and $t = E$;

$$\begin{aligned} f_A(x_i, y_j, z_k) = {} & f_1(x_i, y_j, z_k) - f_2(x_i, y_j, z_k) + f_3(x_i, y_j, z_k) - f_4(x_i, y_j, z_k) \\ & + f_5(x_i, y_j, z_k) - f_6(x_i, y_j, z_k) + f_7(x_i, y_j, z_k) - f_8(x_i, y_j, z_k), \end{aligned} \tag{3.8g}$$

Case 8. $r = O$, $s = O$, and $t = O$;

$$\begin{aligned} f_A(x_i, y_j, z_k) = {} & -f_1(x_i, y_j, z_k) + f_2(x_i, y_j, z_k) - f_3(x_i, y_j, z_k) + f_4(x_i, y_j, z_k) \\ & + f_5(x_i, y_j, z_k) - f_6(x_i, y_j, z_k) + f_7(x_i, y_j, z_k) - f_8(x_i, y_j, z_k), \end{aligned} \tag{3.8h}$$

where f_1,f_2,f_3,f_4,f_5,f_6,f_7, and f_8 refer to the image intensity function defined in the cubes from 1 to 8, respectively. The letters "E" and "O" are the acronyms of even and odd, respectively.

As shown in Figure 1, the removed small cube refers to the first subcube. Therefore, only one-eighth of the whole object space is required to compute the entire set of 3D Zernike moments. The implementation of this property results in 87% reduction in the computational cost. A detailed discussion of this will be found through the following subsections.

3.3. *Fast Computation of Exact 3D Geometric Moments*

General 3D geometric moments of order $(r + s + t)$ for the object intensity function $f(x, y, z)$ are defined as the projection of the function $f(x, y, z)$ onto the monomial $x^r y^s z^t$ as follows:

$$G_{rst} = \iiint_{-\infty}^{\infty} x^r y^s z^t f(x, y, z)\, dx\, dy\, dz. \tag{3.9}$$

Table 2: Numerical values of $x^r\ y^s\ z^t$ are dependent on whatever r, s, and t are even or odd.

r	s	t	$x^r\ y^s\ z^t$							
			P_1	P_2	P_3	P_4	P_5	P_6	P_7	P_8
$E = 2$	$E = 0$	$E = 2$	+0.00156	+0.00156	+0.00156	+0.00156	+0.00156	+0.00156	+0.00156	+0.00156
$E = 0$	$E = 2$	$O = 1$	−0.03107	−0.03107	−0.03107	−0.03107	+0.03107	+0.03107	+0.03107	+0.03107
$E = 2$	$O = 1$	$E = 0$	−0.002071	−0.002071	+0.002071	+0.002071	−0.002071	−0.002071	+0.002071	+0.002071
$E = 2$	$O = 1$	$O = 1$	+0.00092	+0.00092	−0.00092	−0.00092	−0.00092	−0.00092	+0.00092	+0.00092
$O = 1$	$E = 2$	$E = 0$	−0.00621	+0.00621	+0.00621	−0.00621	−0.00621	+0.00621	+0.00621	−0.00621
$O = 1$	$E = 0$	$O = 3$	+0.007629	−0.007629	−0.007629	+0.007629	−0.007629	+0.007629	+0.007629	−0.007629
$O = 1$	$O = 1$	$E = 2$	+0.004577	−0.004577	+0.004577	−0.004577	+0.004577	−0.004577	+0.004577	−0.004577
$O = 1$	$O = 1$	$O = 3$	−0.002023	+0.002023	−0.002023	+0.002023	+0.002023	−0.002023	+0.002023	−0.002023

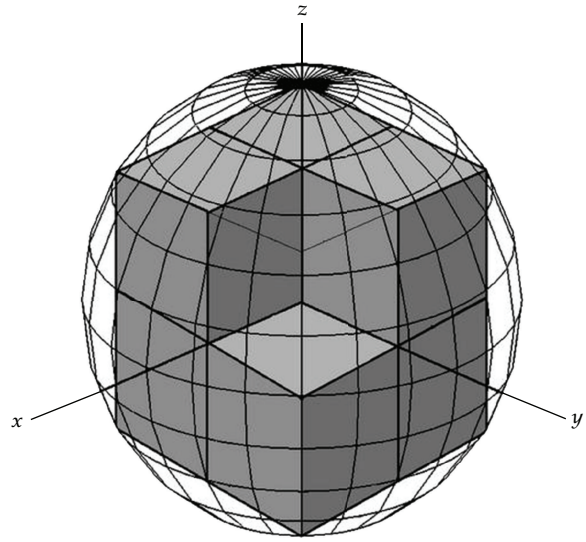

Figure 1: Input 3D image is mapped inside the unit ball.

Based on the circumstances of the present problem, where the input 3D image is mapped inside the unit ball, the upper and lower limits of triple integrals in (3.9) must be rewritten as follows:

$$G_{rst} = \iiint_{-1/\sqrt{3}}^{1/\sqrt{3}} x^r y^s z^t f(x,y,z)\,dx\,dy\,dz. \tag{3.10}$$

Equation (3.10) could be written as follows:

$$G_{pqr} = \sum_{i=1}^{N}\sum_{j=1}^{N}\sum_{k=1}^{N} S_{pqr}(x_i\ , y_j, z_k) f(x_i, y_j, z_k), \tag{3.11}$$

where

$$S_{pqr}(x_i, y_j, z_k) = \int_{x_i-(\Delta x_i/2)}^{x_i+(\Delta x_i/2)} \int_{y_j-(\Delta y_j/2)}^{y_j+(\Delta y_j/2)} \int_{z_k-(\Delta z_k/2)}^{z_k+(\Delta z_k/2)} x^r y^s z^t dx\ dy\ dz. \tag{3.12}$$

The triple integral defined by (3.12) is the source of approximation error. For exact computation of 3D geometric moments, this triple integral could be divided into three separate

individual integrals as follows:

$$I_r(i) = \int_{x_i-(\Delta x_i/2)}^{x_i+(\Delta x_i/2)} x^r dx = \frac{1}{r+1}\left[\left(x_i + \frac{\Delta x_i}{2}\right)^{r+1} - \left(x_i - \frac{\Delta x_i}{2}\right)^{r+1}\right] \tag{3.13a}$$

$$I_s(j) = \int_{y_j-(\Delta y_j/2)}^{y_j+(\Delta y_j/2)} y^s dy = \frac{1}{s+1}\left[\left(y_j + \frac{\Delta y_j}{2}\right)^{s+1} - \left(y_j - \frac{\Delta y_j}{2}\right)^{s+1}\right] \tag{3.13b}$$

$$I_t(k) = \int_{z_t-(\Delta z_t/2)}^{z_t+(\Delta z_t/2)} z^t dz = \frac{1}{t+1}\left[\left(z_t + \frac{\Delta z_t}{2}\right)^{t+1} - \left(z_t - \frac{\Delta z_t}{2}\right)^{t+1}\right]. \tag{3.13c}$$

The upper and lower limits of these integrals are created and stored in a vector form. By applying the symmetry property, the set of geometric moments can thus be computed exactly by

$$G_{pqr} = \sum_{i=1}^{\lfloor N/2\rfloor}\sum_{j=1}^{\lfloor N/2\rfloor}\sum_{k=1}^{\lfloor N/2\rfloor} I_r(i) I_s(j) I_t(k) f_A(x_i, y_j, z_k), \tag{3.14}$$

where $f_A(x_i, y_j, z_k)$ is the augmented intensity function defined by using (3.8a)–(3.8h). For even value of N, the operator $\lfloor N/2\rfloor$ equals to $N/2$, while it equals to $(N-1)/2$ for odd value. The kernel of the exact 3D geometric moments is defined by (3.13a)–(3.13c). This kernel is image-independent. Therefore, this kernel could be precomputed, stored, and recalled whenever it is needed to avoid repetitive computation.

The computational complexity of exact 3D geometric moments could be greatly reduced by applying the successive computation process. The 3D geometric moments of order $(r + s + t)$ defined by (3.14) are computed in three separate steps by successive computation of the 1D sth-order moment for each row, followed by the 2D $(r + s)$th-order moment. Then, the required 3D moment is calculated as the sum of the different 2D moments. This approach was successfully applied in recent works of Hosny [31, 32] where it significantly reduces the total number of required addition and multiplication processes. Equation (3.14) is rewritten in separable forms as follows:

$$G_{rst} = \sum_{k=1}^{\lfloor N/2\rfloor} I_t(z_k) R_{rsk}, \tag{3.15a}$$

where

$$R_{rsk} = \sum_{i=1}^{\lfloor N/2\rfloor} Y_{isk}\ I_r(x_i), \tag{3.16a}$$

$$Y_{isk} = \sum_{j=1}^{\lfloor N/2\rfloor} I_s(y_j) f_A(x_i, y_j, z_k). \tag{3.16b}$$

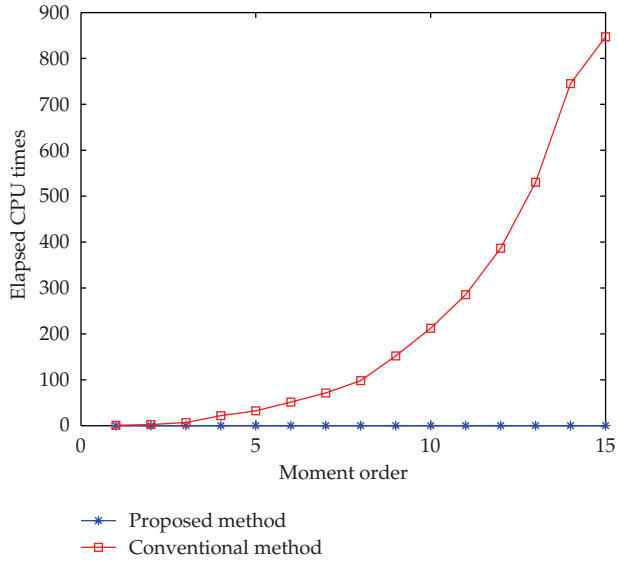

Figure 2: Elapsed CPU times for computing 3D Zernike moments (first numerical experiment).

Table 3: Elapsed CPU times and the reduction percentage for selected moment orders (first numerical experiment).

Moment order	Conventional method [21]	Proposed method	Reduction percentage
1	0.629653	0.057412	90.8820%
3	6.742940	0.078032	98.8428%
5	32.336633	0.095085	99.7060%
10	212.330483	0.174056	99.9180%
12	386.394043	0.182601	99.9527%
15	847.327152	0.276992	99.9673%

4. Experimental Results

The computation of 3D Zernike moments as a liner combination of exact 3D geometric moments ensures the accuracy. This approach was proved through our previously published works of 2D Zernike moments [29, 30]. Based on this fact, this work concentrates on the issue of efficiency in computational time and complexity. The conducted numerical experiments concentrate on the efficiency of the proposed method against the existing methods. The full set of 3D Zernike moments is computed by using the proposed method and the conventional method [21]. Elapsed CPU times for both methods are used to judge the efficiency. Two numerical experiments are conducted. In the first experiment, a randomly generated 3D image with intensity function $f(x_i, y_j, z_k)$ is generated by using the Matlab8 statement $f(x_i, y_j, z_k) = r$ and (M, N, K) where $0 \leq f(x_i, y_j, z_k) \leq 1$ for all i, j, k with $M = N = K = 64$. Selected orders of 3D Zernike moments and the corresponding elapsed CPU times are for both methods shown in Table 3. A graphical representation of these elapsed CPU times is plotted in Figure 2.

In the second experiment, a protein structure of dimensions $70 \times 70 \times 70$ is used. All computational processes are performed by using a code designed with Matlab8 and operated

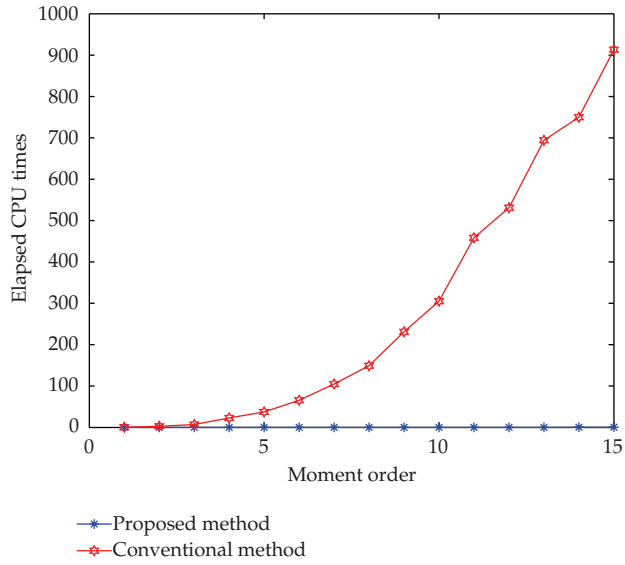

Figure 3: Elapsed CPU times for computing 3D Zernike moments (second numerical experiment).

Table 4: Elapsed CPU times and the reduction percentage for selected moment orders (second numerical experiment).

Moment order	Conventional method [21]	Proposed method	Reduction percentage
1	0.634301	0.068551	89.1927%
3	6.880023	0.080441	98.8308%
5	37.497548	0.132576	99.6464%
10	303.177357	0.198912	99.9344%
12	731.124886	0.282519	99.9614%
15	913.281611	0.342348	99.9625%

on a Lenovo R400 laptop. Similar to the first numerical experiment, selected orders of 3D Zernike moments and their elapsed CPU times for both methods are shown in Table 4. The graphical representation of the elapsed CPU times for moment orders ranging from 1 to 15 is plotted in Figure 3. The logarithmic scale is more suitable to clearly show the big differences in the elapsed CPU times.

It is clear that the proposed method outperformed the conventional method where the reduction in the elapsed CPU times exceeds 95%. In addition to this big reduction, the implementation of symmetry property achieved 87% of memory saving. On the other side, the conventional method does not have any kind of memory saving. Generally, the comparison ensures the superiority of the proposed method.

4.1. Computational Complexity

Complexity analysis of any numerical method presents a simple and clear way to judge the efficiency of this method. Complexity analysis mainly concentrates on the number of operations required by such a method to achieve its goal. The total number of multiplication

and addition operations is the core of the complexity analysis. Evaluation of additional operations such as factorial terms, exponential and power functions is also considered.

The complexity analysis of the proposed and the conventional methods for computing 3D Zernike moments is performed. The computational process of 3D Zernike moments required the computation of 3D geometric moments and the complex coefficients $H_{n,\ell,m}^{rst}$. It is clear that the computation of 3D geometric moments is image-dependent, while the computation of $H_{n,\ell,m}^{rst}$ is image-independent. Therefore, we concentrate on the complexity analysis of 3D geometric moments.

For a 3D digital image of size $N \times N \times N$ and a maximum moment order equal to Max, the complexity analysis of the conventional method represented by (2.16) is discussed first. The total number of arithmetic operations (addition and multiplication) required by the conventional method for computing 3D geometric moments is

$$\frac{(\text{Max}\ +1)(\text{Max}+2)(\text{Max}+3)N^3}{6}, \quad \text{additions} \tag{4.1a}$$

$$\frac{(\text{Max}\ +1)(\text{Max}+2)(\text{Max}+3)N^3}{3}, \quad \text{multiplications.} \tag{4.1b}$$

The corresponding total number of arithmetic operations required by the proposed method will be evaluated. The computational process of the 3D geometric moments by using the proposed method consists of three steps represented by (3.15a), (3.16a), and (3.16b). The total number of arithmetic operations for each step will be discussed individually; then the whole computational complexity will be evaluated. Starting with (3.16b), the creation of the matrix Y_{iqk} requires $(N/2)^3(\text{Max}+1)$ multiplications and $(N/2)^2(N/2-1)(\text{Max}+1)$ additions. The creation process of the matrix R_{pqk} using (3.16a) requires $(N/2)((N/2)-1)(\text{Max}+1)(\text{Max}+2)/2$ additions and $(N/2)^2(\text{Max}+1)(\text{Max}+2)/2$ multiplications. Finally, the computational complexity of the 3D geometric moments using (3.15a) requires $(N/2)(\text{Max}+1)^2(\text{Max}+2)/2$ multiplications and $((N/2)-1)(\text{Max}+1)^2(\text{Max}+2)/2$ additions. Therefore, computing the set of independent exact 3D geometric moments requires the following number of additions and multiplications:

$$\frac{(\text{Max}+1)(\text{Max}+2)}{8}\left[N^2 - 2N + 2\,\text{Max}(N-2)\right] + \frac{N^2(N-2)(\text{Max}+1)}{8}, \quad \text{additions} \tag{4.2a}$$

$$\frac{(\text{Max}+1)(\text{Max}+2)}{8}\left[N^2 + 2N(\text{Max}+1)\right] + \frac{N^3(\text{Max}+1)}{8}, \quad \text{multiplications.} \tag{4.2b}$$

A quick and clear comparison of the complexity of both methods is shown in Table 5 where the total number of arithmetic operations is required by both methods for computing 3D geometric moments with the dimension N and maximum moment order Max. It is clear that the proposed method tremendously reduced the total number of arithmetic operations.

5. Conclusion

This paper proposes very fast and computationally efficient method for computing 3D Zernike moments. In the proposed method, 3D Zernike moments are expressed as a linear

Table 5: Number of addition and multiplication processes required by both methods.

Image size and moment order	Conventional method [21]		Proposed method	
	No. of +	No. of $*$	No. of +	No. of $*$
Max = 5, N = 64	14,680,064	29,360,128	27,063	222,144
Max = 10, N = 64	74,973,184	149,946,368	91,388	451,264
Max = 15, N = 64	213,909,504	427,819,008	206,088	733,184
Max = 20, N = 64	464,257,024	928,514,048	382,788	1,079,904
Max = 5, N = 128	$1.0e + 009 * 0.1174$	$1.0e + 009 * 0.2349$	103,383	1,666,944
Max = 10, N = 128	$1.0e + 009 * 0.5998$	$1.0e + 009 * 1.1996$	329,868	3,200,384
Max = 15, N = 128	$1.0e + 009 * 1.7113$	$1.0e + 009 * 3.4226$	709,128	4,890,624
Max = 20, N = 128	$1.0e + 009 * 3.7141$	$1.0e + 009 * 7.4281$	1,264,788	6,761,664

combination of 3D geometric moments. Numerical experiments show the efficiency of the proposed method, where it achieves more than 95% saving in elapsed CPU times. The implementation of the symmetry property achieves 87% memory saving, which is a very attractive property especially in the processing of 3D images.

Acknowledgments

k. M. Hosny and M. A. Hafez are gratefully acknowledging the financial support of Najran University (Grant no. NU43/2010).

References

[1] M. K. Hu, "Visual pattern recognition by moment invariants," *IRE Transaction on Information Theory*, vol. 8, no. 2, pp. 179–187, 1962.

[2] M. R. Teague, "Image analysis via the general theory of moments," *Journal of the Optical Society of America*, vol. 70, no. 8, pp. 920–930, 1980.

[3] C. H. Teh and R. T. Chin, "On image analysis by the methods of moments," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 10, no. 4, pp. 496–513, 1988.

[4] L. Wang and G. Healey, "Using Zernike moments for the illumination and geometry invariant classification of multispectral texture," *IEEE Transactions on Image Processing*, vol. 7, no. 2, pp. 196–203, 1998.

[5] A. Broumandnia and J. Shanbehzadeh, "Fast Zernike wavelet moments for Farsi character recognition," *Image and Vision Computing*, vol. 25, no. 5, pp. 717–726, 2007.

[6] X. F. Wang, D. S. Huang, J. X. Du, H. Xu, and L. Heutte, "Classification of plant leaf images with complicated background," *Applied Mathematics and Computation*, vol. 205, no. 2, pp. 916–926, 2008.

[7] Y. S. Kim and W. Y. Kim, "Content-based trademark retrieval system using a visually salient feature," *Image and Vision Computing*, vol. 16, no. 12-13, pp. 931–939, 1998.

[8] W. C. Kim, J. Y. Song, S. W. Kim, and S. Park, "Image retrieval model based on weighted visual features determined by relevance feedback," *Information Sciences*, vol. 178, no. 22, pp. 4301–4313, 2008.

[9] S. Li, M. C. Lee, and C. M. Pun, "Complex Zernike moments features for shape-based image retrieval," *IEEE Transactions on Systems, Man, and Cybernetics A*, vol. 39, no. 1, pp. 227–237, 2009.

[10] H. S. Kim and H. K. Lee, "Invariant image watermark using Zernike moments," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 13, no. 8, pp. 766–775, 2003.

[11] Y. Chen, F. Qu, J. Hu, and Z. Chen, "A geometric robust watermarking algorithm based on DWT-DCT and Zernike moments," *Wuhan University Journal of Natural Sciences*, vol. 13, no. 6, pp. 753–758, 2008.

[12] I. A. Ismail, M. A. Shouman, K. M. Hosny, and H. M. Abdel Salam, "Invariant image watermarking using accurate Zernike moments," *Journal of Computer Science*, vol. 6, no. 1, pp. 52–59, 2010.

[13] C. Lu and Z. Lu, "Zernike moment invariants based iris recognition," *Lecture Notes in Computer Science*, vol. 3338, pp. 554–561, 2004.

[14] H. J. Kim and W. Y. Kim, "Eye detection in facial images using Zernike moments with SVM," *ETRI Journal*, vol. 30, no. 2, pp. 335–337, 2008.

[15] V. S. Bharathi and L. Ganesan, "Orthogonal moments based texture analysis of CT liver images," *Pattern Recognition Letters*, vol. 29, no. 13, pp. 1868–1872, 2008.

[16] W. Liyun, L. Hefei, Z. Fuhao, L. Zhengding, and W. Zhendi, "Spermatogonium image recognition using Zernike moments," *Computer Methods and Programs in Biomedicine*, vol. 95, no. 1, pp. 10–22, 2009.

[17] Z. Iscan, Z. Dokur, and T. Ölmez, "Tumor detection by using Zernike moments on segmented magnetic resonance brain images," *Expert Systems with Applications*, vol. 37, no. 3, pp. 2540–2549, 2010.

[18] Q. Ying-Dong, C. Cheng-Song, C. San-Ben, and L. Jin-Quan, "A fast subpixel edge detection method using *Sobel-Zernike moments* operator," *Image and Vision Computing*, vol. 23, no. 1, pp. 11–17, 2005.

[19] T. Funkhouser, P. Min, M. Kazhdan et al., "A search engine for 3D models," *ACM Transactions on Graphics*, vol. 22, no. 1, pp. 83–105, 2003.

[20] N. Canterakis, "3D Zernike moments and Zernike affine invariants for 3D image analysis and recognition," in *Proceedings of the 11th Scandinavian Conference on Image Analysis*, pp. 85–93, 1996.

[21] M. Novotni and R. Klein, "Shape retrieval using 3D Zernike descriptors," *Computer-Aided Design*, vol. 36, no. 11, pp. 1047–1062, 2004.

[22] R. D. Millán, L. Dempere-Marco, J. M. Pozo, J. R. Cebral, and A. F. Frangi, "Morphological characterization of intracranial aneurysms using 3-D moment invariants," *IEEE Transactions on Medical Imaging*, vol. 26, no. 9, pp. 1270–1282, 2007.

[23] W. Qiuting and Y. Bing, "3D terrain matching algorithm and performance analysis based on 3D Zernike moments," in *Proceedings of the International Conference on Computer Science and Software Engineering (CSSE '08)*, vol. 6, pp. 73–76, Wuhan, China, December 2008.

[24] L. Sael, D. La, B. Li et al., "Rapid comparison of properties on protein surface," *Proteins*, vol. 73, no. 1, pp. 1–10, 2008.

[25] L. Sael, B. Li, D. La et al., "Fast protein tertiary structure retrieval based on global surface shape similarity," *Proteins*, vol. 72, no. 4, pp. 1259–1273, 2008.

[26] G. A. Papakostas, Y. S. Boutalis, D. A. Karras, and B. G. Mertzios, "A new class of Zernike moments for computer vision applications," *Information Sciences*, vol. 177, no. 13, pp. 2802–2819, 2007.

[27] M. S. Al-Rawi, "Fast Zernike moments," *Journal of Real-Time Image Processing*, vol. 3, no. 1-2, pp. 89–96, 2008.

[28] C.-Y. Wee and R. Paramesran, "On the computational aspects of Zernike moments," *Image and Vision Computing*, vol. 25, no. 6, pp. 967–980, 2007.

[29] K. M. Hosny, "Fast computation of accurate Zernike moments," *Journal of Real-Time Image Processing*, vol. 3, no. 1-2, pp. 97–107, 2008.

[30] K. M. Hosny, "A systematic method for efficient computation of full and subsets Zernike moments," *Information Sciences*, vol. 180, no. 11, pp. 2299–2313, 2010.

[31] K. M. Hosny, "Exact and fast computation of geometric moments for gray level images," *Applied Mathematics and Computation*, vol. 189, no. 2, pp. 1214–1222, 2007.

[32] K. M. Hosny, "Fast and low-complexity method for exact computation of 3D Legendre moments," *Pattern Recognition Letters*, vol. 32, no. 9, pp. 1305–1314, 2011.