Fast and accurate method for radial moment's computation

Khalid M. Hosny *

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

A R T I C L E   I N F O

Article history:
Received 30 November 2008
Received in revised form 7 July 2009
Available online 3 October 2009
Communicated by J.A. Robinson

Keywords:
Radial moments
Geometric moments
Exact computation
Circularly moments
Symmetry property

A B S T R A C T

Fast and accurate method is proposed for radial moment's computation. Exact radial moments are computed as a linear combination of exact geometric moments. The digital image is transformed to be inside the unit circle, where the transformed image is divided into four quadrants. Based on the symmetry property; only one quadrant of transformed image is needed to compute the whole set of moments. This leads to significant reduction in the computational complexity requirements. The proposed method completely removes the approximation errors and tremendously reduced the computational demands. Numerical experiments are performed, where the obtained results are compared with the approximated values. The obtained results clearly explained the efficiency of the proposed method.

© 2009 Elsevier B.V. All rights reserved.

1. Introduction

Circularly orthogonal moments, such as Zernike, Pseudo-Zernike and Fourier-Mellin moments are generally used to represent an image with the minimum amount of information redundancy (Teague, 1980). In addition to this attractive property, the set of circularly orthogonal moments are rotation and flipping invariants by nature, while the translation and scale invariants are easily achieved through the normalization of their polynomials. Based on their attractive characteristics, circularly orthogonal moments are widely used in image processing, pattern recognition and computer vision. Circularly orthogonal moments are used as invariant pattern or object recognition (Khotanzad and Hong, 1990; Wang and Healey, 1998; Kan and Srinath, 2002; Broumandnia and Shanbehzadeh, 2007), content-based image retrieval (Kim and Kim, 1998), watermarking and data-hiding (Kim and Lee, 2003; Xin et al., 2004; Amin and Subhulukshini, 2004), edge detection (Ghosal and Mehrotra, 1992; Dong et al., 2005; Bin et al., 2008), image segmentation (Ghosal and Mehrotra, 1993), biomedical engineering (Iskander et al., 2001, 2002), medical imaging (Bharathi and Ganesan, 2008), and face recognition (Haddadnia et al., 2003; Kim and Kim, 2008; Kanan and Faez, 2009).

Radial moment's computation is the main part of the computational process of these orthogonal moments, where all of these moments could be expressed as linear combinations of radial moments. Approximate computation of radial moments produced numerical instabilities and consequently degraded the quality of the computed descriptors.

Digital images are usually defined in the Cartesian coordinates while the circular moments are by nature defined in the polar coordinates. Consequently, computation of these circular moments required square-to-circle transformation which produced what is called geometric error. The other kind of error is the numerical error which is the direct result of approximation process. Zernike (ZMs) and Pseudo-Zernike moments (PZMs) could be expressed as a linear combination of radial or geometric moments of the same order or less (Teh and Chin, 1988). Orthogonal Fourier-Mellin moments (OFMMs) were defined first by Sheng and Shen (1994). Similar to ZMs and PZMs, OFMMs could be expressed as a linear combination of radial or geometric moments. It is clear that, the computational accuracy of all the aforementioned circularly orthogonal moments is dependent on the computational accuracy of the radial moments.

Recently Wee and Paramesran (2006), proposed a method that compute approximate radial moments using symmetry property. In fact, their method reduces the computational complexity requirements but on the other side it produces a set of approximate radial moments, where the numerical error problem is still unsolved.

Apart from the circular orthogonal moments, other orthogonal moments like Legendre, Gegenbauer, Tchebichef and Krawtchouk moments could be expressed as a linear combination of only geometric moments of the same order or less. Also Novotni and Klein (2004) shows that 3D Zernike moments could be expressed as a combination of geometric moments. Therefore, the implementation...
of symmetrical property to radial moment’s computation as done by Wee and Paramesran (2006) limited its benefit to circular orthogonal moments, while, its implementation to geometric moment’s computation make it useful in the computation of all orthogonal moments and their extension to three-dimension. This is the motivation of this work, where the symmetrical property is implemented in the process of exact geometric moment’s computation.

This paper proposes a new method for accurate computation of radial moments for gray-level images and objects. The radial moments are computed exactly as a linear combination of exact geometric moments, while the later are computed exactly by using a mathematical integration of the monomial polynomials. The symmetry property and fast algorithm are applied for computational complexity reduction. Experimental results clearly show the efficiency of this proposed method.

The rest of the paper is organized as follows: in Section 2, an overview of the radial moments is presented. The proposed method is described in Section 3. Section 4 is devoted to numerical experiments. Conclusion and concluding remarks are presented in Section 5.

2. Radial moments

Radial or rotational moments of order \( p \) and repetition \( q \) are defined as:

\[
R_{pq} = \int_0^{2\pi} \int_0^1 r^p e^{-ip\theta} f(r \cos \theta, r \sin \theta) r dr d\theta. \tag{1}
\]

where \( i = \sqrt{-1}, p = 0, 1, 2, 3, \ldots \infty \) and \( q \) is any positive or negative integer. Based on Eq. (1), radial moments are defined in terms of polar coordinates \((r, \theta)\) over a unit disk. On the other side, image intensity function defined in Cartesian coordinates \((x, y)\). Consequently, an appropriate image mapping is imperative. There are mainly two traditional mapping approaches. In the first approach, the square image plane is mapped onto a unit disk, where the center of the image is assumed to be the origin of coordinates. In this approach, all pixels outside the unit disk are ignored, which results in a loss of some image information. In the second approach, the whole square image is mapped inside the unit disk, where the center of the image is assumed to be the coordinate origin. The second approach overcomes the lost information problem in the first kind of transformation.

For a digital image of size \( N \times N \) the integrals in Eq. (1) are replaced by summations and the image is normalized inside the unit disk using second aforementioned mapping transformations. The approximated radial moments are:

\[
\tilde{R}_{pq} = i_p \sum_{l=-\infty}^{\infty} \sum_{j=-\infty}^{\infty} p^l e^{-ipl} f(l, j). \tag{2}
\]

Eq. (2) is so-called direct method for radial moment’s computation, which is the approximated version using zeroth-order approximation (ZOA). \( i_p \) is the total number of pixels that achieve the condition \( |l| \leq 1 \). This equation has two sources of errors, the first one is the numerical error and the other is the geometrical error. The numerical error is caused by approximating integrals in Eq. (1) through replacing them by summations. Based on the principles of mathematical analysis, summations are equivalent to integrals as the number of sampling points tends to infinity. Consequently, the numerical error increases as the number of sampling points decreases. Also, this error increases as the order of moments increases. Therefore, numerical instabilities are faced when the moment order reaches a certain value. The geometrical error is caused by a square to circular mapping transformation.

2.1. Circular orthogonal moments via radial moments

Circular orthogonal moments are represented as a linear combination of radial moments of the same order or less. Relations of ZMs, PZMs and OFMMs with radial moments are briefly discussed through the following subsections.

2.1.1. Zernike moments

The complex two-dimensional Zernike moments of order \( p \) and repetition \( q \) are defined as a linear combination of radial moments as follows:

\[
Z_{pq} = \frac{p+1}{\pi} \sum_{k=0}^{p} \sum_{p-k \text{ even}} B_{pk} R_{p0}, \tag{3}
\]

where \( p = 0, 1, 2, 3, \ldots \infty \) and \( q \) is positive integer according to the conditions \( p - q = \text{even} \), \( q \leq p \). Zernike moments with negative values of repetition \( q \) are obtained directly by making use of the complex conjugate of Zernike moments in Eq. (3). The coefficient matrix \( B_{pk} \) is defined as:

\[
B_{pk} = (-1)^{\frac{p}{2}} \binom{\frac{p}{2}}{\frac{k}{2}} ! i^{k-q} R_{p0}. \tag{4}
\]

and recursively computed through the following relations:

\[
B_{pp} = 1, \tag{5.1}
\]

\[
B_{pq-2ip} = \frac{p+q}{p-q+2} B_{pp}, \tag{5.2}
\]

\[
B_{pq(k-2)} = -\frac{(p+k)(p-q)}{(p+k)(p-k+2)} B_{pk}. \tag{5.3}
\]

2.1.2. Pseudo-Zernike moments

Pseudo-Zernike moments of order \( p \) and repetition \( q \) are defined as a linear combination of radial moments as follows:

\[
A_{pq} = \frac{p+1}{\pi} \sum_{k=-q}^{p} C_{pk} R_{p0}, \tag{6}
\]

where the coefficient matrix \( C_{pk} \) is defined as:

\[
C_{pk} = \frac{(-1)^{p-k}(p+k+1)!}{(p-k)!(p-q+1)!(p-k+q)!}. \tag{7}
\]

Similar to the previous case, this matrix could be computed through the following recurrence relations:

\[
C_{pp} = 1, \tag{8.1}
\]

\[
C_{p(q+k)} = \frac{k+q+1}{k-q+1} C_{pk}, \tag{8.2}
\]

\[
C_{p(q-k)} = \frac{(p+k)!(p-k)!}{(p-k+1)!(p-k+q)!} B_{pk}. \tag{8.3}
\]

2.1.3. Fourier-Mellin moments

Orthogonal Fourier-Mellin moments of order \( p \) and repetition \( q \) are defined as a linear combination of radial moments as follows:

\[
O_{pq} = \frac{p+1}{\pi} \sum_{k=-q}^{p} O_{pk} R_{p0}, \tag{9}
\]

where the coefficient matrix \( O_{pk} \) is defined as:

\[
O_{pk} = \frac{(-1)^{p-k}(p+k+1)!}{(p-k)!}(k+1)! . \tag{10}
\]
This matrix is computed using the following equations:
\[
\alpha_{ij} = (-1)^j (p + 1),
\]
\[
\alpha_{jk} = \frac{(p + k + 1)(p - k + 1)}{k(k + 1)} \alpha_{j(k-1)}.
\]

3. The proposed method

The proposed method aims to provide a fast computation of exact radial moments. In addition to these elegant characteristics, the proposed method is a low-complexity method where it reduces the requirements by 75%. Through the next subsection, all these characteristics will be discussed in details.

3.1. Symmetry property

A digital image of size \(N \times N\) is an array of pixels. The 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 \((x_i, y_j) \in [0, N - 1] \times [0, N - 1]\). \(\Delta x_i = x_{i+1} - x_i\) \(\Delta y_j = y_{j+1} - y_j\) are sampling intervals in the \(x\)- and \(y\)-directions, respectively. The second square-to-circle mapping approach is applied as shown in Fig. 1, where the transformed coordinates are:
\[
\begin{align*}
\xi_i &= \frac{2j - N - 1}{N\sqrt{2}}, & \eta_j &= -\frac{2j - N - 1}{N\sqrt{2}}, \\
\nu_{ij} &= \sqrt{(\xi_i)^2 + (\eta_j)^2}, & \theta_{ij} &= \tan^{-1} \left( \frac{\eta_j}{\xi_i} \right)
\end{align*}
\]

with \(i = 1, 2, \ldots, N\) and \(j = 1, 2, \ldots, N\).

The transformed image is inside the unit circle. The center of this image coincides with the Cartesian coordinate origin. Both axes divide the transformed image into four quadrants. Each point \(P_1\) with the Cartesian coordinates \((x_i, y_j)\) in the first quadrant which has three similar points in the other three quadrants as shown in Fig. 2. These points are \(P_2(x_N-i+1, y_j), P_3(x_N-i+1, y_{N-j+1})\) and \(P_4(x_i, y_{N-j+1})\). All of these four points have the same radial distance from the origin point as shown in Fig. 3.

The geometric moments of the order \((p + q)\) are the projection of the image function \(f(x, y)\) onto the monomial \(x^p y^q\) and defined as:
\[
M_{pq} = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} x^p y^q f(x, y) \, dx \, dy.
\]

Since the points \(P_1, P_2, P_3\) and \(P_4\) has the same radial distance, then; the numerical value of \(x^p y^q\) will be dependent on whatever \(p\) and \(q\) are even or odd. For more clarification, we consider the following illustrative example. Assume the first point \(P_1\) has the coordinates \(P_1(x_1, y_1) = P_1(5/8\sqrt{2}, 3/8\sqrt{2})\). Consequently, the coordinates of the other three points are \(P_2(x_N-i+1, y_j), P_3(x_N-i+1, y_{N-j+1})\) and \(P_4(x_i, y_{N-j+1})\). Numerical values of \(x^p y^q\) for the points \(P_1, P_2, P_3\) and \(P_4\) with different possibilities of exponent indices \(p\) and \(q\) are listed in Table 1.

Based on this symmetry property and the results obtained in Table 1, the geometric moments can then be evaluated according to the following four cases:

\[
\begin{align*}
\text{Case 1:} & \quad p \text{ and } q \text{ are both even; } \\
& \quad \hat{f}_x(x, y) = \hat{f}_x(x, y) + \hat{f}_x(x, y) + \hat{f}_x(x, y) + \hat{f}_x(x, y). \\
\text{Case 2:} & \quad p \text{ is even and } q \text{ is odd; } \\
& \quad \hat{f}_x(x, y) = \hat{f}_x(x, y) + \hat{f}_x(x, y) - \hat{f}_x(x, y) - \hat{f}_x(x, y). \\
\text{Case 3:} & \quad p \text{ is odd and } q \text{ is even; } \\
& \quad \hat{f}_x(x, y) = \hat{f}_x(x, y) - \hat{f}_x(x, y) - \hat{f}_x(x, y) + \hat{f}_x(x, y). \\
\text{Case 4:} & \quad p \text{ and } q \text{ are both odd; } \\
& \quad \hat{f}_x(x, y) = \hat{f}_x(x, y) - \hat{f}_x(x, y) + \hat{f}_x(x, y) - \hat{f}_x(x, y).
\end{align*}
\]
where \( f(x_i, y_j) \) is the intensity function of the pixel point \((x_i, y_j)\) in the first quadrant; the other functions \( f_3(x_i, y_j), f_4(x_i, y_j) \) and \( f_5(x_i, y_j) \) are the intensity functions at the corresponding pixel points in the second, third and fourth quadrants, respectively.

### 3.2. Exact computation of radial moments

According to the square-to-circle transformation, the transformed image is defined in the square \([-\sqrt{2}, 1/\sqrt{2}] \times [-\sqrt{2}, 1/\sqrt{2}]\); therefore, the \((p+q)\) order geometric moments are defined as:

\[
M_{pq} = \int \frac{1}{\sqrt{2}} \int \frac{1}{\sqrt{2}} x^p y^q f(x, y) dx dy. \tag{15}
\]

Radial moments are expressed as a linear combination of geometric moments of the same order or less as follows (Hosny, 2008):

\[
R_{pq} = \sum_{m=0}^{p} \sum_{m=0}^{q} w^m \left( \frac{N}{2} \right) \left( \frac{q}{m} \right) M_{p-2m-q, m} \tag{16.1}
\]

with

\[
R_{pp} = \sum_{m=0}^{p} w^m \left( \frac{p}{m} \right) M_{p-m, m} \tag{16.2}
\]

where \( S(p - q)/2, w = -\sqrt{-1} \) if \( p > 0 \) or \( w = -\sqrt{-1} \) if \( q \leq 0 \). Based on Eq. (16), exact computation of geometric moments results in exact values of radial moments. As shown in (Hosny, 2008), the time-consuming direct computations of factorial terms are avoided by using recurrence relations:

\[
D_{p,0} = 1. \tag{17.1}
\]

\[
D_{p,0} = 1. \tag{17.2}
\]

\[
D_{p,k} = \frac{p}{p-k} D_{p-1,k}. \tag{17.3}
\]

\[
D_{p,k} = \frac{1}{k(p-k)} D_{p,k-1}. \tag{17.4}
\]

It is clear that, the matrix \( D \) is independent on the image, where its dimensions are dependent only on the moment’s order. Therefore, this matrix is pre-computed and stored for future use.

Similar to our previous work (Hosny, 2007), exact geometric moments for the whole input image could be easily obtained through the computation of the first quadrant only. This could be achieved by using the augmented intensity function \( f_0(x_i, y_j) \) defined in Eq. (14). As discussed in the previous section, the augmented intensity function has four different values based on whatever the indices \( p \) and \( q \) are even or odd as follows:

\[
\hat{M}_{pq} = \sum_{i=1}^{N} I_p(i) I_q(j) f_k(x_i, y_j), \tag{18}
\]

where

\[
\left\lfloor \frac{N}{2} \right\rfloor = \left\lfloor (N-1)/2 \right\rfloor. \quad N \text{ is odd},
\]

\[
\left\lfloor \frac{N}{2} \right\rfloor = \left\lfloor N/2 \right\rfloor. \quad N \text{ is even}.
\]

Implementation of Eq. (18) results in the reduction of the computational cost by 75%, where only one quadrant is considered. For more details about computational complexity, the reader is advised to read Section 4.2. The kernels \( I_p(i) \) and \( I_q(j) \) are defined as follows:

\[
I_p(i) = \frac{1}{p+1} \left[ U_{p+1}^{i+1} - U_{p+1}^{i} \right], \tag{20.1}
\]

\[
I_q(j) = \frac{1}{q+1} \left[ V_{q+1}^{j+1} - V_{q+1}^{j} \right]. \tag{20.2}
\]

with

\[
U_{i+1} = x_i + \frac{\Delta x}{2}, \tag{21.1}
\]

\[
U_i = x_i - \frac{\Delta x}{2}, \tag{21.2}
\]

\[
V_{j+1} = y_j + \frac{\Delta y}{2}, \tag{21.3}
\]

\[
V_j = y_j - \frac{\Delta y}{2}. \tag{21.4}
\]

The time complexity of Eq. (18) could be significantly reduced by successive computation of the 1D qth order moments for each row. Eq. (18) will be rewritten in the following separable form:

\[
\hat{M}_{pq} = \sum_{i=1}^{N} I_p(i) Y_{iq}, \tag{22}
\]

where

\[
Y_{iq} = \sum_{j=1}^{N} I_q(j) f_k(x_i, y_j). \tag{23}
\]

\( Y_{iq} \) in Eq. (23) is the qth order moment of row \( i \). Since,

\[
I_0(i) = \sqrt{2}/N.
\]

Substitute Eq. (24) into Eq. (18), yields:

\[
\hat{M}_{pq} = \sqrt{2/N} \sum_{i=1}^{N} Y_{iq}. \tag{25}
\]
4. Numerical experiments

In this section, the validity proof of the proposed method will be presented. Artificial test images are used in our numerical experiments where, radial moments that are computed using the proposed method are compared with theoretical and ZOA approximate values. Both ZOA and Wee’s method (Wee and Parmesran, 2006), are used to approximately compute a set of radial moments, where the later one was derived from the same formula of ZOA method. The artificial test images are relatively small so that hand calculations can be easily employed. CPU elapsed time for real standard images are used to compare the required computational time of the proposed method against ZOA and Wee’s method.

4.1. Artificial test images

4.1.1. First image

Artificial test images of small size are used to prove validity of the proposed method, where hand calculations could be employed and the theoretical values easily obtained. A special image with intensity function, \( f(x, y) = 1 \) for all points \((x, y)\) is considered. The size of this artificial test images is \( 4 \times 4 \). The original image that is defined in the square \([-1/\sqrt{2},1/\sqrt{2}] \times [-1/\sqrt{2},1/\sqrt{2}]\). Consequently, the set of two-dimensional geometric moments of order \((p + q)\) are:

\[
M_{pq} = \frac{(1/\sqrt{2})^{p+1} - (1/\sqrt{2})^{q+1}}{p+1} (1/\sqrt{2})^{q+1} - (1/\sqrt{2})^{q+1},
\]

\[p+1\]

Eq. (26) can be simplified as follows:

\[
M_{pq} = \frac{4(1/\sqrt{2})^{p+2} - (p+1)(q+1)}{(p+1)(q+1)}, \quad p = \text{even},
\]

\[p = \text{odd}.
\]

In this case, theoretical values of radial moments are calculated by using Eq. (27) into Eq. (16). The corresponding exact values are calculated using Eqs. (22)–(25) and (14) in Eq. (16). The ZOA approximated values using Eq. (2). It is clear that exact and theoretical values are identical. For quick comparison, all calculated values are shown in Table 2.

4.1.2. Second image

The intensity function of the second artificial test image is represented by the matrix: \( A = [3, 2, 1, 5, 6, 1, 7, 3, 2, 8, 4, 6, 5, 1, 4, 2] \). Radial moments for this image are shown in Table 3. It is obvious that the computed values using the proposed method are identical to theoretical values, while the approximate ZOA values deviate from the theoretical values especially when the moment order increases, see Table 3.

4.2. Computational complexity

Complexity analysis of the considered methods is very important, where such analysis give a simple and clear way to judge the efficiency of the different methods. Complexity analysis mainly concentrates on the number of multiplications and additions required by each method. Evaluation of factorial terms, exponential and power functions are considered if encounter in any one of the considered methods. For a gray-level image of size \( N \times N \) and a maximum moment order equal to Max, the analysis of the direct ZOA approximation method represented by Eq. (2) is discussed first. Computation of an individual radial moment required the evaluation of the power function \( r_i^p \) and the exponential function \( e^{-q_{ij}} \) plus six multiplication process. Three are included in the evaluation of the exponential function and the rest are a result of multiplying both power and exponential function with the image

| Table 2 |
| Table 3 |
exponential functions. The method of (Wee and Paramesran, 2006) is very similar to the direct method except the complexity analysis of geometric moments’ computations methods: for gray-level image of size $N \times N$ and a maximum moment order equal to $\text{Max}$.

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>$6N^2(\text{Max} + 1)^2$</td>
<td>$6(N/2)^2(\text{Max} + 1)^2$</td>
<td>$N^2(\text{Max} + 1)^2/2$</td>
<td>$N^2(\text{Max} + 1)^2/2$</td>
</tr>
<tr>
<td>$N^2(\text{Max} + 1)^2/2$</td>
<td>$(N/2)^3(\text{Max} + 1)^2/2$</td>
<td>$(N/2)^3(\text{Max} + 1)^2/2$</td>
<td>$(N/2)^3(\text{Max} + 1)^2/2$</td>
</tr>
<tr>
<td>$N^2(\text{Max} + 1)^2$</td>
<td>$(N/2)^2(\text{Max} + 1)^2$</td>
<td>$(N/2)^2(\text{Max} + 1)^2$</td>
<td>$(N/2)^2(\text{Max} + 1)^2$</td>
</tr>
<tr>
<td>Exponential functions</td>
<td>$N^2(\text{Max} + 1)^2$</td>
<td>$(N/2)^2(\text{Max} + 1)^2$</td>
<td>$(N/2)^2(\text{Max} + 1)^2$</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>GM</th>
<th>Multiplications</th>
<th>Additions</th>
</tr>
</thead>
<tbody>
<tr>
<td>ZOA method</td>
<td>$(\text{Max} + 1)(\text{Max} + 2)N^2$</td>
<td>$(\text{Max} + 1)(\text{Max} + 2)/2$</td>
</tr>
<tr>
<td>Wee’s method (2006)</td>
<td>$(\text{Max} + 1)(\text{Max} + 2)(\text{Max} + 2)/2$</td>
<td>$(\text{Max} + 1)(\text{Max} + 2)/2$</td>
</tr>
<tr>
<td>Hosny’s method (2007)</td>
<td>$(\text{Max} + 1)(N^2 + \text{Max} + 1)$</td>
<td>$(\text{Max} + 1)(N^2 + \text{Max} + 1)/2$</td>
</tr>
<tr>
<td>Proposed method</td>
<td>$(\text{Max} + 1)(N^2 + \text{Max} + 4)$</td>
<td>$(\text{Max} + 1)(N^2 + \text{Max} + 4)/2$</td>
</tr>
</tbody>
</table>

Fig. 4. Test images: (a) peppers and (b) boat.
quick comparison of these four methods could be done for \( N = 512 \)
and \( \text{Max} = 55 \), the direct ZOA (Wee et al., 2008; Hosny, 2007) and
the proposed method require 836763648, 209190912, 16257080
and 4064312 multiplications, respectively. These methods require
418381824, 104595456, 15465408 and 4062660 additions, respec-
tively. It is clear that, the proposed method is the most efficient
one.

4.3. Computational time

The main drawback of using circularly moments is their con-
ssuming computational time. Therefore, computational time reduc-
tion is a very important issue especially for large size images and
objects. The CPU elapsed time is used in the comparison process
in all the performed numerical experiments. All our numerical
experiments are performed with 1.8 GHz Pentium IV PC with 512
MBYTE RAM. The executed code is designed by using MATLAB7.
The set of radial moments is computed by using the proposed
method, the method of (Wee and Paramesran, 2006) and the
approximation ZOA method. In the first experiment, a gray-scale
image of peppers with size \( 128 \times 128 \) as in Fig. 4a is used. The
CPU elapsed times for the three different methods are included
in Table 6. It is clear that, the proposed method reduces the execution
time tremendously.

In the second numerical experiment, the set of radial moments
are computed for the boat test images with size \( 512 \times 512 \). This
test image is relatively large. The CPU elapsed times are graphically
shown in Fig. 5. Tables 7 shows the elapsed times of the different
methods for a selected moment orders. Similar to results of the
previous numerical experiments the execution time of ZOA and
Wee’s methods monotonically increase as the moment order in-
crease, while the execution time required by the proposed method
is extremely small. Based on the obtained results we conclude that,
both ZOA and Wee’s methods are impractical for large images and
moments with higher orders.

The obtained results in this section clearly shows that, our pro-
posed method is accurate where the set of radial moments are
computed exactly using a set of exact geometric moments, while,
on the other side, Wee’s method is inaccurate where the set of ra-
dial moments are computed approximately. The comparison of the
CPU elapsed times ensures the superiority of the proposed method
against Wee’s method, where the proposed method extremely re-
duces the execution times.

5. Conclusion

This work proposes fast, accurate and low-complexity method
of radial moment computation for gray-scale images. Due to the
wide range of their applications, the accurate computation of circu-
lar moments is a very crucial problem. Based on their relation with
the geometric moments, radial moments are computed exactly.
Consequently, the set of circular moments could be accurately
computed based on their representation as a linear combination of
radial moments. The implementation of the symmetry property
significantly reduces the computational complexity demands. The
numerical experiments are performed for real images with differ-
ent sizes to ensure the efficiency of the proposed method.

### Table 6

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Max = 5</td>
<td>0.7340</td>
<td>0.203000</td>
<td>0.016000</td>
</tr>
<tr>
<td>Max = 15</td>
<td>7.4370</td>
<td>2.047000</td>
<td>0.047000</td>
</tr>
<tr>
<td>Max = 25</td>
<td>23.6100</td>
<td>5.703000</td>
<td>0.110000</td>
</tr>
<tr>
<td>Max = 35</td>
<td>53.0310</td>
<td>11.875000</td>
<td>0.219000</td>
</tr>
<tr>
<td>Max = 45</td>
<td>91.9690</td>
<td>22.344000</td>
<td>0.453000</td>
</tr>
<tr>
<td>Max = 55</td>
<td>137.3600</td>
<td>36.063000</td>
<td>0.860080</td>
</tr>
<tr>
<td>Max = 65</td>
<td>193.4380</td>
<td>51.985000</td>
<td>1.231000</td>
</tr>
</tbody>
</table>

### Table 7

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Max = 5</td>
<td>14.0620</td>
<td>4.141000</td>
<td>0.859000</td>
</tr>
<tr>
<td>Max = 15</td>
<td>177.5310</td>
<td>44.922000</td>
<td>1.312000</td>
</tr>
<tr>
<td>Max = 25</td>
<td>501.5780</td>
<td>141.407000</td>
<td>1.797000</td>
</tr>
<tr>
<td>Max = 35</td>
<td>991.7030</td>
<td>278.187000</td>
<td>2.438000</td>
</tr>
<tr>
<td>Max = 45</td>
<td>1596.9680</td>
<td>455.672000</td>
<td>3.219000</td>
</tr>
<tr>
<td>Max = 55</td>
<td>2452.4530</td>
<td>690.047000</td>
<td>4.328000</td>
</tr>
</tbody>
</table>

![Fig. 5. Linear scale of CPU elapsed time in seconds for the 512 × 512 gray-scale boat image](image-url)