Regular paper

A New Convolution Model for the Fast Computation of Zernike Moments

Chandan Singh, Ashutosh Aggarwal, Sukhjeet Kaur Ranade, Chandan Singh

PII: S1434-8411(16)30305-3
DOI: http://dx.doi.org/10.1016/j.aeue.2016.11.014
Reference: AEUE 51730

To appear in: *International Journal of Electronics and Communications*

Received Date: 25 June 2016
Revised Date: 12 November 2016
Accepted Date: 18 November 2016

Please cite this article as: C. Singh, A. Aggarwal, S. Kaur Ranade, C. Singh, A New Convolution Model for the Fast Computation of Zernike Moments, *International Journal of Electronics and Communications* (2016), doi: http://dx.doi.org/10.1016/j.aeue.2016.11.014

Title: A New Convolution Model for the Fast Computation of Zernike Moments

Author Name: **Chandan Singh (Ph.D.)**
Professor, Department of Computer Science, Punjabi University, Patiala-147002, India
E-mail: chandan.csp@gmail.com, Phone: 91-175-3046316, M: 91-9872043209

Author Name: **Ashutosh Aggarwal (M.Tech.)**
Research Scholar, Department of Computer Science, Punjabi University, Patiala-147002, India
E-mail: er.ashutoshaggarwal@gmail.com

Author Name: **Sukhjeet Kaur Ranade (Ph.D.)**
Assistant Professor, Department of Computer Science, Punjabi University, Patiala-147002, India
E-mail: mrsdjjr@gmail.com

Corresponding Author Name: **Chandan Singh (Ph.D.)**
Professor, Department of Computer Science, Punjabi University, Patiala-147002, India
E-mail: chandan.csp@gmail.com, Phone: 91-175-3046316, M: 91-9872043209

A New Convolution Model for the Fast Computation of Zernike Moments

Chandan Singh, Ashutosh Aggarwal, and Sukhjeet Kaur Ranade

Abstract

Zernike moments (ZMs) are very useful image descriptors which belong to a family of orthogonal rotation invariant moments. Due to their many attractive characteristics, they have been used in many pattern recognition, image processing and computer vision applications. However, they suffer from very high computation complexity which prohibits their use in many practical problems. The ZMs are computed as a convolution process between the image data and the Zernike kernel functions. In the past, various attempts have been made for the efficient computation of ZMs and considerable success has been achieved using recursive relations and 8-way symmetry/anti-symmetry of Zernike function. In this paper, we propose a new computational flow model for the convolution of the image data with the Zernike kernel functions. The proposed model also takes advantage of the 8-way symmetry/anti-symmetry property of the kernel function and builds up the convolution process which reduces the number of additions/subtractions from 56 to 24 and the number of multiplications from 12 to 8 (refer Table 6 in the text) for each location in an octant of a circular disk on which the moments are computed. Detailed experimental results show that the speed of the ZMs computation increases by a factor varying from 15% to 41% (depending upon the order of moments) for multiple images as compared to the existing fast algorithms available in the literature. When ZMs are computed at each pixel of an image on overlapping blocks, the improvement in computation time varies from 10% to 33%.

Keywords

Zernike moments; fast convolution; orthogonal moments; rotation invariance

1. Introduction

Introduced by Teague in 1980 to image analysis [1], Zernike moments (ZMs) belong to a family of orthogonal rotation invariant moments (ORIMs) which have been found very useful for many pattern recognition, computer vision and image processing applications. Being orthogonal, they possess minimum information redundancy and a set of low order moments are sufficient to describe images uniquely. The magnitude of ZMs is rotation invariant. They are computed over a unit disk and hence they are scale invariant after mapping a discrete image into a unit disk. If the center of the unit disk is placed at the center of the mass of the image, it becomes translation invariant. The moments are computed using an integration process and hence they are resilient to image noise. Because of their many attractive characteristics, ZMs have been extensively used in vast range of applications. The use of ZMs in any particular application is determined by the number of ZMs used in that application. The low order ZMs mainly represents the global shape of the object while the high order ZMs describes the subtle details in an object. Therefore, both low and high order ZMs find its use in different set of applications. The low order ZMs are mainly used in applications including optical character recognition [2,3], object recognition and pattern classification [4-8], content based image retrieval [9-12], 2D/3D object recognition through sketches [13,14], estimation of rotation angle between images and determination of image symmetry [15-17], image denoising [18], image super-resolution [19], etc. While, the high order moments are mainly used in biometric studies where identification and classification of human characteristics involves subtle and complex patterns. Therefore, high order ZMs are used in palmprint verification [20], recognition of facial information [21], hand shape verification [22], etc. Apart from biometric identification, high order ZMs are used in applications like image watermarking [23-25] and image reconstruction [26-29].

One of the major issues involved in the computation of ZMs is its high computation complexity. The computation of the ZMs involves two parts. The first part pertains to the computation of the kernel function at each pixel and the second part relates to the convolution process of the pixel values with the kernel function. The computation of the kernel function is very time consuming. If all moments upto a maximum

order p_{max} are computed for an image of size $N \times N$, then the time complexity of the kernel function is $O(N^2 p_{max}^3)$ using the conventional method. There exist fast algorithms which reduce the computation complexity to $O(N^2 p_{max}^2)$ [30-34]. The focus of these approaches is to reduce the computation complexity of Zernike polynomials (ZPs) from $O(p_{max}^3)$ to $O(p_{max}^2)$, when ZPs of all orders and repetitions are required. The basic strategy of these approaches is to use recursive relations to derive a higher order polynomial from its low order polynomials. Singh and Walia [33,34] observed that these algorithms are not only fast but numerically more stable for high order of moments. The numerical stability is achieved because the recursive methods do not involve the explicit computation of the factorial terms which cause numerical instability when factorials of large integer numbers are computed. Typically, it is shown in [35] that factorial of an integer equal to or greater than 21 is computed incorrectly by using double precision (digital machines have limitations of handling factorial of integers larger than 12 using 32-bit integer arithmetic [35]). It may be noted that the moments whether computed by the classical methods involving factorial terms or by the fast recursive methods which do not involve the factorial terms yield the same values for low order of moments. The deviation occurs when factorial of a number more than 21 is involved. Further, enhancement in the speed of computation of ZMs is achieved by using 8-way symmetry/ anti-symmetry property of the radial and angular kernel functions [36,37], which can save the computation time upto 87% approximately. Recently, Upneja and Singh [38] have derived recursive relations for the computations of Jacobi-Fourier moments of which ZMs is a particular case.

The efforts made in the last two decades for the fast computation of the kernel function of ZMs have reached to a saturation point. The use of the recursive relations for the computation of the radial and angular parts of the kernel function and their 8-way symmetry/anti-symmetry property results in significant reduction in the computation time. The ZMs upto order $p_{max} = 20$ for a grayscale image of size 512×512 can be computed in less than $0.40\ seconds$ on a machine with 3.00 GHz CPU, 512 MB RAM using Visual C++ 6.0 under the Microsoft Windows environment [34]. The ZMs are computed as a convolution

process between the image data and the Zernike kernel function. The Zernike kernel function is separable in radial and angular parts which possess 8-way symmetry and anti-symmetry property. Therefore, once the kernel function is computed, the time taken by the convolution process forms the major portion in the overall time computation of the ZMs. In this paper, we develop a new computational flow model which enhances the speed of the existing convolution process between the image data and the kernel function of the ZMs. The proposed fast convolution process reduces the number of additions/subtractions from 56 to 24 and the number of multiplications from 12 to 8 as compared to the existing method [36,37], per pixel location on the octant of a circular disk. The motivation for this idea stems from the fast computation of discrete cosine transform [39]. Although the method has been derived for ZMs, the proposed method is also applicable for other orthogonal and non-orthogonal rotation invariant moments and transforms such as the rotational moments, pseudo-Zernike moments (PZMs), orthogonal Fourier-Mellin moments (OFMMs), angular radial transforms (ART), and polar harmonic transforms (PHTs). Zhu et al. [40] have recently developed a series of unit disc-based generalized orthogonal moments (DGMs) of which ZMs, PZMs, and OFMMs are special cases. They have shown that their proposed descriptors are superior to the classical disc-based moments in terms of image representation capability and classification accuracy. The proposed fast method for the computation of the ZMs is also applicable to the fast computation of DGMs.

The rest of the paper is organized as follows. An overview of the ZMs is given in Section 2. For the sake of the completeness of the proposed method, the existing fast algorithms using recursive relations for the computation of radial and angular kernel functions, and the use of 8-way symmetry/anti-symmetry property is given in Section 3. Section 4 presents the proposed fast convolution process for the fast computation of ZMs which reduces the number of operations from 56 additions/subtractions and 12 multiplications to 24 additions/subtractions and 8 multiplications, respectively. The experimental results are presented in Section 5 and Section 6 concludes the paper.

2. The Zernike Moments (ZMs)

The ZMs of order p with repetition q of an image function $f(x,y)$ in two dimensions over a unit disk are defined by:

$$Z_{pq} = \frac{p+1}{\pi} \iint_{x^2+y^2 \le 1} f(x,y)\, V_{pq}^{*}(x,y)\, dx\, dy \tag{1}$$

where p is a non-negative integer, q is an integer such that $|q| \le p, and\ p - |q| = even.$ The function $V_{pq}^{*}(x,y)$ is the complex conjugate of the Zernike polynomials (ZPs) $V_{pq}(x,y)$ defined as:

$$V_{pq}(x,y) = V_{pq}(r,\theta) = R_{pq}(r) e^{jq\theta} \tag{2}$$

where

$$r = \sqrt{x^2+y^2}, \qquad j = \sqrt{-1}, \qquad \theta = \tan^{-1}\left(y/x\right), \qquad \theta \in [0, 2\pi] \tag{3}$$

The function $R_{pq}(r)$ is the radial polynomial given by

$$R_{pq}(r) = \sum_{s=0}^{(p-|q|)/2} \frac{(-1)^s (p-s)!}{s!\left(\frac{p+|q|}{2}-s\right)!\left(\frac{p-|q|}{2}-s\right)!}\, r^{p-2s} \tag{4}$$

The set of ZPs $\{V_{pq}(x,y)\}$ are orthogonal and complete over the unit disk:

$$\iint_{x^2+y^2 \le 1} V_{pq}^{*}(x,y) V_{p'q'}(x,y)\, dx\, dy = \frac{\pi}{p+1} \delta_{pp'} \delta_{qq'} \tag{5}$$

where δ_{pq} is Kronecker delta. The orthogonality and completeness property of ZMs provides the image reconstruction capability. The ZMs are continuous moments and hence they provide infinite number of the moment coefficients Z_{pq}. If all moments upto a maximum order p_{max} are given, then the image function can be reconstructed:

$$\hat{f}(x,y) = \sum_{p=0}^{p_{max}} \sum_{q=-p}^{p} Z_{pq}\, V_{pq}(x,y), \quad p - |q| = even \tag{6}$$

The higher is the value of p_{max}, the closer will be $\hat{f}(x,y)$ to $f(x,y)$.

2.1. ZMs for digital images

In digital image processing, an image function $f(x,y)$ is defined over a two-dimensional grid. Let $f(i,k)$ be a rectangular arrangement of pixels of size $N \times N$, then the zeroth order approximation of Eq. (1) results in [41,42]:

$$Z_{pq} = \frac{(p+1)}{\pi} \sum_{i=0}^{N-1} \sum_{k=0}^{N-1} f(i,k)\, V_{pq}^{*}(x_i, y_k) \Delta x_i \Delta y_k$$
$$= \frac{4(p+1)}{\pi D^2} \sum_{i=0}^{N-1} \sum_{k=0}^{N-1} f(i,k)\, V_{pq}^{*}(x_i, y_k) \tag{7}$$

where

$$x_i = \frac{2i+1-N}{D}, \qquad y_k = \frac{2k+1-N}{D}, \qquad \Delta x_i = \Delta y_k = \frac{2}{D}, \qquad i,k = 0,1,\dots,N-1 \tag{8}$$

$$D = \begin{cases} N, & \textit{for inner disk mapping} \\ N\sqrt{2}, & \textit{for outer disk mapping} \end{cases} \tag{9}$$

The mapping defined by Eq. (8) converts the domain of a digital image $f(i,k)$ of size $N \times N$ into a unit disk [41,42].

Equation (7) is used for the computation of ZMs for an entire image. Here, the size of kernels is same as the size of the image. However, there are certain applications such as image denoising [18] and image super-resolution [19] where ZMs are computed over image blocks of size $n \times n$, where n varies from 7 to 11. In such applications, for an image having $M \times N$ pixels, we need to compute ZMs over $n \times n$ size blocks at MN locations. Thus, computing kernels once for a block of size $n \times n$ and performing proposed fast convolution process over the entire image will save significant amount of time. Therefore, when ZMs have to be computed at each pixel location of an image over a block (or sub-image) of size $n \times n$ (say, $n = 7$), then the zeroth order approximation can be expressed in the form of conventional convolution:

$$Z_{pq}(x,y) = \frac{4(p+1)}{\pi D^2} \sum_{i=0}^{n-1} \sum_{k=0}^{n-1} f(x+i, y+k) V_{pq}^{*}(i,k), \tag{10}$$
$$x = 0,1,2,\dots,M-1, \qquad y = 0,1,2,\dots,N-1$$

The kernels $V_{pq}^{*}(i,k)$ are computed once and saved in a table, and the indices x_i, y_k used in Eq. (7) are replaced by (i,k) to make them independent of x and y. Also, in Eq. (10) $D = n$ or $n\sqrt{2}$ according to Eq. (9).

3. Existing Methods for Fast Computations of ZMs

The existing methods for the fast computation of ZMs are based on three strategies: i) recursive relations for the computation of the radial polynomials which reduce the order of complexity of a polynomial from $O(p)$ to $O(1)$ where p is the degree of the polynomial, ii) recursive relations for the angular polynomials, $e^{-jq\theta}$, which avoid the direct computation of the trigonometric functions $\cos(q\theta)$, and $\sin(q\theta)$, and iii) the 8-way symmetry/anti-symmetry property of the ZPs. These approaches are discussed in detail in [32-34]. Here, we present the overview of these methods in order to elaborate the background of our work.

3.1. Recursive relations for the computation of radial polynomials

The time complexity of computing ZMs for all orders upto p_{max} is $O(N^2 p_{max}^3)\big(= O(N^2 p_{max}^2 \times p_{max})\big)$, where the factor $N^2 p_{max}^2$ arises due to the total number of pixels (N^2) and total number of moments represented by p_{max}^2. The second factor p_{max} represents the time complexity of the radial part of the Zernike polynomial given by Eq. (4). The order of complexity assumes very high values when the size of the image (N^2) and p_{max} become very large. Therefore, attempts are made to reduce the time taken to compute the radial polynomials and bring it to a constant order $O(1)$. For this process, three forms of the recurrence relations have been proposed which reduce the complexity of a polynomial of a degree p from $O(p)$ to $O(1)$. These methods are known as Kintner's method [30], Prata's method [31], and q-recursive method [32]. These methods have further been investigated in depth by [32-35].

3.2. Recursive computation of angular functions

The trigonometric functions $cos(q\theta)$ and $sin(q\theta)$ present in the angular function $e^{-jq\theta}$ can be computed recursively as follows.

$$\cos(q\theta) = 2\cos\theta\cos((q-1)\theta) - \cos((q-2)\theta) \tag{11}$$

$$\sin(q\theta) = 2\cos\theta\sin((q-1)\theta) - \sin((q-2)\theta) \tag{12}$$

with

$$\cos(0) = 1, \quad \sin(0) = 0, \quad \cos\theta = x_i/r_{ik}, \quad \sin\theta = y_k/r_{ik}, \quad r_{ik} = \sqrt{x_i^2 + y_k^2},$$

where (x_i, y_k) are the Cartesian coordinates of the pixel (i,k) defined by Eq. (8). These relations avoid the direct computation of $cos(q\theta)$ and $sin(q\theta)$ which are computationally very expensive.

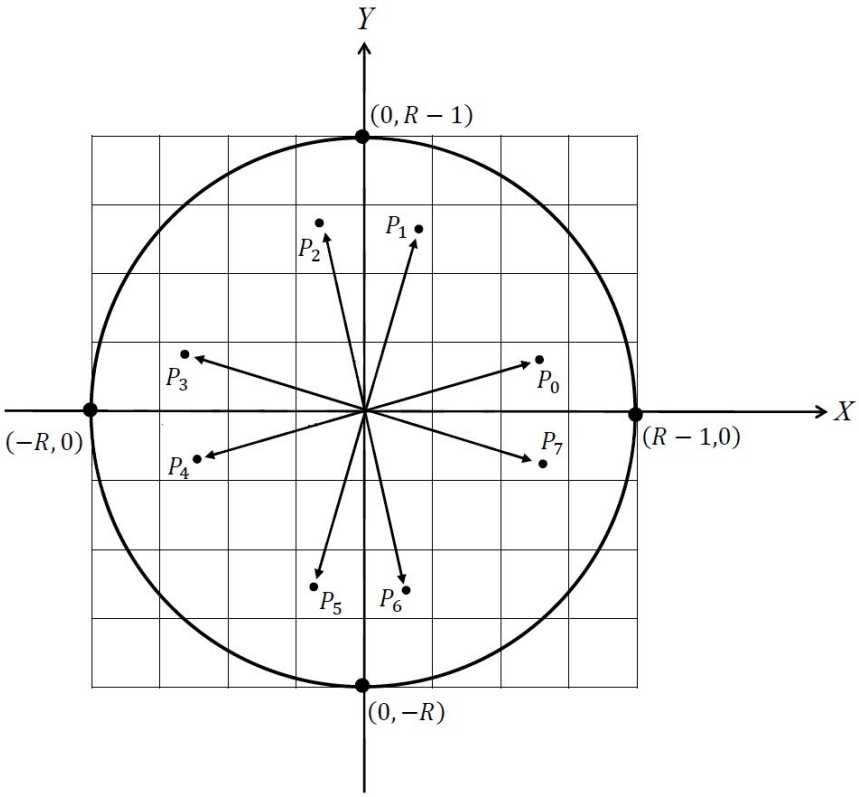
3.3. 8-Way Symmetry/Anti-Symmetry Property

The real and imaginary parts of the ZMs can be written explicitly from Eq. (7):

$$Z_{pq} = \frac{4(p+1)}{\pi D^2}\sum_{i=0}^{N-1}\sum_{k=0}^{N-1} f(i,k)\,R_{pq}(r_{ik})[\cos(q\theta_{ik}) - j\sin(q\theta_{ik})] \tag{13}$$

where $\cos(q\theta_{ik})$, $\sin(q\theta_{ik})$, and r_{ik} are computed as given in Eqs. (11) and (12).

Let P_0 be the location of the pixel (i,k) in the first octant of a circular disk of digital radius R ($R = \lfloor N/2 \rfloor$ in the present case) as shown in Fig. 1, then the pixels on the other seven octants at locations $P_1 = (k,i)$, $P_2 = (-k-1,i)$, $P_3 = (-i-1,k)$, $P_4 = (-i-1,-k-1)$, $P_5 = (-k-1,-i-1)$, $P_6 = (k,-i-1)$, and $P_7 = (i,-k-1)$ will have the same radial distance r_{ik} from the center of the unit disk. If the origin of an image of size $N \times N$ is located at the top-left corner of the image, then the absolute locations of these points with respect to the center of the image will be $(R+i, R+k)$, $(R+k, R+i)$, $(R-k-1, R+i)$, ..., $(R+i, R-k-1)$, respectively. The values of $cos(q\theta)$ and $sin(q\theta)$ at these locations are depicted in Tables 1 and 2 as derived in [43]. It is obvious from these tables that $cos(q\theta)$ and $sin(q\theta)$ are computed only once for the location P_0 and then these values are used at seven other locations P_1 through P_7.



$P_0 = (i, k),$	$P_1 = (k, i),$
$P_2 = (-k - 1, i),$	$P_3 = (-i - 1, k),$
$P_4 = (-i - 1, -k - 1)$	$P_5 = (-k - 1, -i - 1)$
$P_6 = (k, -i - 1)$	$P_7 = (i, -k - 1)$

Fig. 1. 8-way symmetry of a pixel $P_0 = (i, k)$. The origin of the coordinate system is at the center of the image.

Using 8-way symmetry/anti-symmetry, Eq. (13) is converted into the following form (refer [43] for detail)

$$Z_{pq} = \frac{4(p+1)}{\pi D^2} \sum_{i=0}^{R-1} \sum_{k=0}^{i} [(A + B) - j(C + D)] \tag{14}$$

where expressions for A, B, C, and D are defined in Table 3. The function values f_0 through f_7 are pixel values at locations P_0 through P_7, respectively. More precisely, $f_0 = f(R + i, R + k)$, $f_1 = f(R + k, R + i)$, $f_2 = f(R - k - 1, R + i)$, $f_3 = f(R - i - 1, R + k)$, $f_4 = f(R - i - 1, R - k - 1)$, $f_5 = f(R - k - 1, R - i - 1)$, $f_6 = f(R + k, R - i - 1)$, and $f_7 = f(R + i, R - k - 1)$. Also, given in Table 3 are $VR = Re\left(V_{pq}^*(r_{ik}, \theta_{ik})\right) = R_{pq}(r_{ik}) \cos(q\theta_{ik})$ and $VI = Im\left(V_{pq}^*(r_{ik}, \theta_{ik})\right) = -R_{pq}(r_{ik}) \sin(q\theta_{ik})$, where

$Re(z)$ and $Im(z)$ represent the real and imaginary parts of a complex number z. It is observed that $R_{pq}(r_{ik})$, $\cos(q\theta_{ik})$, and $\sin(q\theta_{ik})$ are computed only once at the pixel location (i,k) and Eq. (14) is derived by using their 8-way symmetry/anti-symmetry property. It is worth mentioning here that when the size of an image is even, the pixels on the positive x-axis, y-axis and on the diagonal of the first quadrant have 4-way symmetry because of the digital nature of the image. For these cases the values of $A, B, C, and\ D$ are obtained by assuming $f_1 = f_3 = f_5 = f_7 = 0$. When the size of an image is odd, we have one more special case pertaining to the central pixel which lies on the center of the disk. For such a case we set $f_i = 0, i = 1,2, \dots ,7$.

Table 1.Values of $\cos(q\theta)$ at eight symmetric/anti-symmetric locations P_0 through P_7

$mod(q,4)$	Values of $\cos(q\theta)$ at locations P_0 to P_7							
	P_0 $q\theta$	P_1 $q\left(\frac{\pi}{2}-\theta\right)$	P_2 $q\left(\frac{\pi}{2}+\theta\right)$	P_3 $q(\pi-\theta)$	P_4 $q(\pi+\theta)$	P_5 $q\left(\frac{3\pi}{2}-\theta\right)$	P_6 $q\left(\frac{3\pi}{2}+\theta\right)$	P_7 $q(2\pi-\theta)$
0	$\cos(q\theta)$	$\cos(q\theta)$	$\cos(q\theta)$	$\cos(q\theta)$	$\cos(q\theta)$	$\cos(q\theta)$	$\cos(q\theta)$	$\cos(q\theta)$
1	$\cos(q\theta)$	$\sin(q\theta)$	$-\sin(q\theta)$	$-\cos(q\theta)$	$-\cos(q\theta)$	$-\sin(q\theta)$	$\sin(q\theta)$	$\cos(q\theta)$
2	$\cos(q\theta)$	$-\cos(q\theta)$	$-\cos(q\theta)$	$\cos(q\theta)$	$\cos(q\theta)$	$-\cos(q\theta)$	$-\cos(q\theta)$	$\cos(q\theta)$
3	$\cos(q\theta)$	$-\sin(q\theta)$	$\sin(q\theta)$	$-\cos(q\theta)$	$-\cos(q\theta)$	$\sin(q\theta)$	$-\sin(q\theta)$	$\cos(q\theta)$

Table 2.Values of $\sin(q\theta)$ at eight symmetric/anti-symmetric locations P_0 through P_7

$mod(q,4)$	Values of $\sin(q\theta)$ at locations P_0 to P_7							
	P_0 $q\theta$	P_1 $q\left(\frac{\pi}{2}-\theta\right)$	P_2 $q\left(\frac{\pi}{2}+\theta\right)$	P_3 $q(\pi-\theta)$	P_4 $q(\pi+\theta)$	P_5 $q\left(\frac{3\pi}{2}-\theta\right)$	P_6 $q\left(\frac{3\pi}{2}+\theta\right)$	P_7 $q(2\pi-\theta)$
0	$\sin(q\theta)$	$-\sin(q\theta)$	$\sin(q\theta)$	$-\sin(q\theta)$	$\sin(q\theta)$	$-\sin(q\theta)$	$\sin(q\theta)$	$-\sin(q\theta)$
1	$\sin(q\theta)$	$\cos(q\theta)$	$\cos(q\theta)$	$\sin(q\theta)$	$-\sin(q\theta)$	$-\cos(q\theta)$	$-\cos(q\theta)$	$-\sin(q\theta)$
2	$\sin(q\theta)$	$\sin(q\theta)$	$-\sin(q\theta)$	$-\sin(q\theta)$	$\sin(q\theta)$	$\sin(q\theta)$	$-\sin(q\theta)$	$-\sin(q\theta)$
3	$\sin(q\theta)$	$-\cos(q\theta)$	$-\cos(q\theta)$	$\sin(q\theta)$	$-\sin(q\theta)$	$\cos(q\theta)$	$\cos(q\theta)$	$-\sin(q\theta)$

Table 3. Clustering of image pixels at 8-radially symmetrical locations for the computation of Z_{pq} as given by Eq. (14), where $VR = Re\left(V_{pq}^{*}(r_{ik}, \theta_{ik})\right), VI = Im\left(V_{pq}^{*}(r_{ik}, \theta_{ik})\right), Re(z) and\ Im(z)$ are the real and imaginary parts of the complex number z

$mod(q,4)$	A	B	C	D
0	$\begin{pmatrix} f_0 + f_1 + f_2 + f_3 + f_4 \\ + f_5 + f_6 + f_7 \end{pmatrix} * VR$	0	0	$\begin{pmatrix} f_0 - f_1 + f_2 - f_3 + f_4 \\ - f_5 + f_6 - f_7 \end{pmatrix} * VI$
1	$(f_0 - f_3 - f_4 + f_7) * VR$	$(f_1 - f_2 - f_5 + f_6) * VI$	$(f_1 + f_2 - f_5 - f_6) * VR$	$(f_0 + f_3 - f_4 - f_7) * VI$
2	$\begin{pmatrix} f_0 - f_1 - f_2 + f_3 + f_4 \\ - f_5 - f_6 + f_7 \end{pmatrix} * VR$	0	0	$\begin{pmatrix} f_0 + f_1 - f_2 - f_3 + f_4 \\ + f_5 - f_6 - f_7 \end{pmatrix} * VI$
3	$(f_0 - f_3 - f_4 + f_7) * VR$	$(-f_1 + f_2 + f_5 - f_6) * VI$	$(-f_1 - f_2 + f_5 + f_6) * VR$	$(f_0 + f_3 - f_4 - f_7) * VI$

All previous papers on the fast computation of ZMs use recurrence relations for the computation of $R_{pq}(r_{ik}), \cos(q\theta_{ik})$, and $\sin(q\theta_{ik})$ and use 8-way symmetry/anti-symmetry property using Eq. (14). After applying these strategies, it is observed from Eq. (14) that the major part of the computation time is taken for the evaluation of the expressions contained in $A, B, C, and\ D$. The conventional way of computing A, B, C, D requires 52 additions and 12 multiplications, as can be observed from Table 3. However, in the following section, we develop a fast convolution scheme which reduces the number of additions/subtractions from 52 to 20 and number of multiplications from 12 to 8, involved in the computation of $A, B, C, and\ D$. Therefore, for overall convolution process the number of additions/subtractions and multiplications are reduced from 56 to 24 and 12 to 8, respectively.

4. A New Convolution Model for the Fast Computation of ZMs

The computation of ZMs involves two parts: 1) Computation of Zernike kernel functions (real and imaginary parts of Zernike polynomials) and 2) Convolution of the Zernike kernel functions with the image data. The existing fast algorithms focus on the fast computation of the Zernike kernel function $V_{pq}^{*}(r, \theta)$ and achieve significant improvement in their computation efficiency (i.e. part 1 of ZMs computation). The ZMs, however, are obtained as a result of the convolution of the image function $f(i, k)$ with the kernel function $V_{pq}^{*}(r, \theta)$. The 8-way symmetry/anti-symmetry property of the kernel function enables the grouping of the pixel values which are reflected in the expressions contained in the variables

$A, B, C, and\ D$. The major computation time is taken by the kernel function because of its high time complexity. However, when we have to process a number of images with fixed size, the kernel function is computed only once because it is independent of the image data. In such applications, each image is convolved with the fixed kernel and ZMs are computed as a result of the convolution operation. Therefore, a procedure which efficiently computes the expressions represented by $A, B, C, and\ D$ in Table 3 will be much more useful in such practical applications. Following the scheme presented in [39] for the fast computation of 1-D DCT, we present the fast computation of the expressions $A, B, C, and\ D$ in four stages. In other words, we have proposed a new convolution model that speeds up the part 2 of ZMs computation, keeping part 1 similar to existing methods. Figure 2 represents computational flow diagram of the proposed convolution model depicting the four stages designed for the fast computations of Zernike moments. The building blocks (or symbols) used in the computational flow diagram are explained in Fig. 3 in terms of their mathematical equations as well as the number of operations required to execute them.

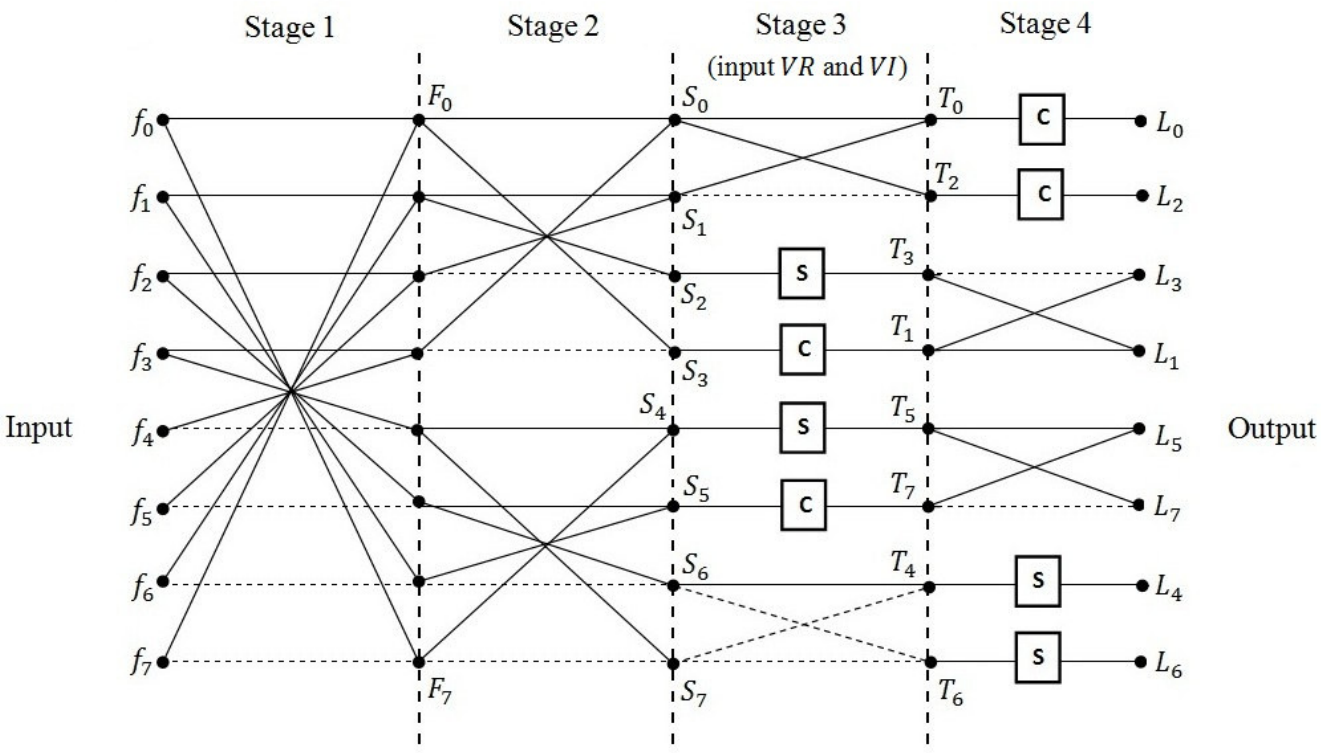


Fig. 2. Computational flow diagram for the fast convolution of the image data with the kernel functions of ZMs.

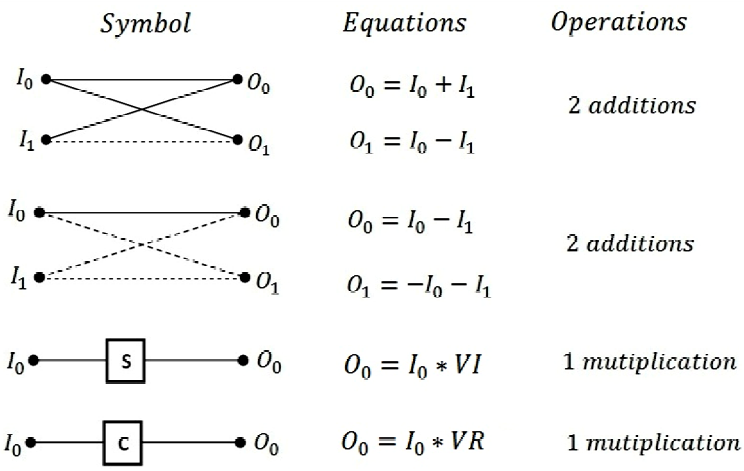


Fig. 3. Symbols used in computational flow diagram for fast computations of Zernike moments.

Table 4 represents all the operations involved in each stage of the computational flow diagram and the expressions involved for the calculation of the intermediate variables $F_i, S_i, and\ T_i$, and finally the output L_i from the input values $f_i, i = 0,1, \dots, 7$. At each stage, values in Table 4 are shown in the same order in which they appear in the computational flow diagram (refer Fig. 2). The convolution process starts at stage 3. At this stage, the terms $T_1, T_3, T_5, and\ T_7$ are derived after convolving the results of stage 2 with either the real part VR $\left(VR = R_{pq}(r_{ik})\cos(q\theta_{ik})\right)$ or the imaginary part VI $\left(VI = -R_{pq}(r_{ik})\sin(q\theta_{ik})\right)$ of the kernel function $V^*_{pq}(r_{ik}, \theta_{ik})$. The rest of the terms, which are not convolved at stage 3, are convolved at stage 4.

Table 4. Tabular representation of the operations in the proposed convolution model as shown in Fig.2

$VR = R_{pq}(r_{ik})\cos(q\theta_{ik}), VI = -R_{pq}(r_{ik})\sin(q\theta_{ik})$

Input	Stage 1	Stage 2	Stage 3 (input VR and VI)	Stage 4
f_0	$F_0 = f_0 + f_7$	$S_0 = F_0 + F_3$	$T_0 = S_0 + S_1$	$L_0 = T_0 * VR$
f_1	$F_1 = f_1 + f_6$	$S_1 = F_1 + F_2$	$T_2 = S_0 - S_1$	$L_2 = T_2 * VR$
f_2	$F_2 = f_2 + f_5$	$S_2 = F_1 - F_2$	$T_3 = S_2 * VI$	$L_3 = T_1 - T_3$
f_3	$F_3 = f_3 + f_4$	$S_3 = F_0 - F_3$	$T_1 = S_3 * VR$	$L_1 = T_1 + T_3$
f_4	$F_4 = f_3 - f_4$	$S_4 = F_4 + F_7$	$T_5 = S_4 * VI$	$L_5 = T_5 + T_7$
f_5	$F_5 = f_2 - f_5$	$S_5 = F_5 + F_6$	$T_7 = S_5 * VR$	$L_7 = T_5 - T_7$
f_6	$F_6 = f_1 - f_6$	$S_6 = F_5 - F_6$	$T_4 = S_6 - S_7$	$L_4 = T_4 * VI$
f_7	$F_7 = f_0 - f_7$	$S_7 = F_4 - F_7$	$T_6 = -(S_6 + S_7)$	$L_6 = T_6 * VI$

Table 5. Values of the expressions $A, B, C, and\ D$ using the proposed convolution model as used in Eq. (14)

$mod(q,4)$	A	B	C	D	$A+B$	$C+D$
0	L_0	0	0	L_4	L_0	L_4
1	T_1	T_3	T_7	T_5	L_1	L_5
2	L_2	0	0	L_6	L_2	L_6
3	T_1	$-T_3$	$-T_7$	T_5	L_3	L_7

Table 5 presents the values of the expressions $A, B, C, and\ D$ using the proposed fast convolution model as used in Eq. (14), in terms of the variables defined in Table 4. It can be observed from Table 3 that the computation of $A, B, C, and\ D$ involves 52 additions/subtractions and 12 multiplications. On the other hand, the proposed fast convolution model requires only 20 additions/subtractions and 8 multiplications for their computation. Thus, from a total of 64 operations, only 28 operations are required. Therefore, even if some computational overheads such as storage and access of the intermediate values $F_i, S_i, T_i, and\ L_i, i = 0,1, \dots, 7$ are added, which are very small as compared to the arithmetic operations: addition and multiplication, we expect significant improvement in the computation time because of the reduction in the number of arithmetic operations. Table 6 shows the comparison between the existing convolution approaches [36,37] and the proposed fast convolution approach in terms of the number of operations required for the overall convolution process and for the computation of the expressions $A, B, C, and\ D$. It can be seen from the table that the proposed convolution model reduces the number of operations by 53-56% approximately over the existing convolution approaches [36,37], which in turn would reduce the time complexity of ZMs computation and thereby, making it more efficient than the existing fast approaches [34].

Table 6. Comparison between the existing convolution approaches [36,37] and the proposed fast convolution approach in terms of the number of operations performed

Number of operations performed for	Existing convolution approaches [36,37]	Proposed fast convolution approach	% reduction in operations $\frac{(\text{Existing} - \text{Proposed})}{\text{Existing}} \times 100$
Computation of $A, B, C, and\ D$	64 (52 additions/subtractions and 12 multiplications)	28 (20 additions/subtractions and 8 multiplications)	56%
Overall convolution process	68 (56* additions/subtractions and 12 multiplications)	32 (24* additions/subtractions and 8 multiplications)	53%

*Additional 4 additions required in A+B and C+D in Eq. (14).

The four stages of the computational flow diagram (Fig. 2) have to be executed in sequence due to the data dependencies that exists between them. However, calculation inside one stage can be parallelized. For example, during stage 2, the computation of $S_0, S_1, S_2, and\ S_3$ is independent from the computation of $S_4, S_5, S_6, and\ S_7$. Hence, both of these computations can be parallelized to reduce the time taken for ZMs computation. Similar independencies in computations are also found in the other stages of the computational flow diagram, which can be further exploited to minimize the overall computation time.

5. Experimental Results

The computation of ZMs comprises two parts. The first part is the computation of the ZMs kernel functions $V_{pq}^{*}(r, \theta)$ and the second part is the convolution of these kernel functions with the image data $f(i, k)$. In this paper, a new convolution flow model is proposed for fast computation of ZMs. The proposed convolution model reduces the computation time by decreasing the number of additions/subtractions from 56 to 24 and the number of multiplications from 12 to 8 required for the convolution operation in the second part. To demonstrate the effectiveness of the proposed convolution model, two sets of experiments have been performed as follows.

1. First set of experiments highlights the usefulness of the proposed convolution model in applications like object recognition [4], image retrieval [9], etc., where the key requirement is the fast computation of

ZMs from multiple images, so as to reduce the overall response time of the system. Since ZMs kernel functions are independent of image data, the ZMs kernel functions are precomputed only once and are stored in a table. Thus, the overall response time in such applications is highly affected by the computation time of the convolution process. Since the proposed convolution model significantly reduces the number of operations in the convolution process, the time needed for the convolution process will reduce and, in turn, the overall time taken to compute ZMs will reduce. Hence, the proposed convolution model for the fast computation of ZMs would prove to be very useful in such applications.

2. Second set of experiments highlights the usefulness of the proposed convolution model in applications like image denoising [18], image super-resolution [19], etc., where for a given (single) image we need to compute ZMs over image blocks (sub-images) of size $n \times n$ (where, n varies from 7 to 11), at each pixel location. Thus, computing kernels once for a block of size $n \times n$ and performing proposed fast convolution process over the entire image will save significant amount of time.

The experimental results obtained by the proposed convolution model are compared with the convolution process employed by [37] and with Algorithm E in [34]. All three methods under consideration use the 8-way symmetry/anti-symmetry property for fast computation of ZMs. It is worth noting that Algorithm E in [34] is the fastest method used for the computation of ZMs among all fast methods available in the literature. This method is not only the fastest but also provides the most stable numerical values for high order ZMs. Another important aspect to note is that the accuracy of the computed moments is not affected because of the proposed fast convolution model. The proposed method does not change other aspects of the ZMs such as accuracy, noise immunity or image reconstruction, which remain the same. It focuses only on the fast computation of ZMs through the proposed fast convolution process. Thus, it is possible to provide accurately reconstructed images of any size from higher orders of ZMs with reduced computation time. The

proposed convolution model and other compared methods are implemented in Microsoft Visual C++ 6.0 under Microsoft Windows 7 on a PC with 2.13 GHz CPU and 3 GB RAM.

5.1. Fast Computation of ZMs Using The Proposed Convolution Model For Multiple Images

In this set of experiments we present the average computation time required to compute ZMs form 12 standard gray scale images [41] which are widely used in various image processing applications. The computation time depends on the image size and the maximum number of moments p_{max} and is independent of image contents. The experiments are performed on images with two different sizes, 256×256 and 512×512 pixels. The value of maximum number of moments p_{max} is varied from order 5 to order 50 with equal increments of five. Tables 7 and 8 display the results of the average computation time t_{avg} versus the maximum number of moments $p_{max} (p_{max} \in \{5,10,\dots,50\})$ for images of size 256×256 and 512×512 using the convolution process of [37], Algorithm E of [34], and the proposed convolution model. It is observed that the proposed approach yields an improvement in the computation speed varying from 19.58% to 40.11% over the convolution process of [37] and from 15.34% to 28.47% over Algorithm E of [34], for the images of size 256×256. The percentage enhancement in computational speed for images of size 512×512 varies from 20.81% to 41.34% and from 17.56% to 31.86% over the convolution process of [37] and Algorithm E of [34], respectively. It is also observed here that the higher is the moment order, the better is the improvement in speed. This is due to the fact that the proposed approach computes the values $F_i, S_i, T_i, and\ L_i, i = 0,1,\dots,7$ (refer Tables 4 and 5) only once for all moment orders upto p_{max} and then uses these values for the computation of all ZMs of all orders upto p_{max}. Therefore, the repetitive operations on the data are avoided. In order to present the qualitative improvement in speed, the graphs for the average CPU elapse time versus p_{max} for two different image sizes are shown in Figs. 4-5.

Table 7. Average CPU elapsed time (sec) versus the maximum order of moments p_{max} for the computation of ZMs for the proposed method, method of [37] and Algorithm E of [34] on 12 different images of size 256×256

Maximum order of ZMs (p_{max})	**Method of [37] (A)**	**Algorithm E of [34] (B)**	**Proposed method (C)**	**% improvement $\frac{(A-C)}{A} \times 100$**	**% improvement $\frac{(B-C)}{B} \times 100$**
5	0.048	0.045	0.038	19.58%	15.34%
10	0.143	0.133	0.109	23.63%	17.75%
15	0.286	0.261	0.209	26.96%	20.01%
20	0.481	0.433	0.338	29.83%	22.13%
25	0.723	0.643	0.489	32.37%	23.87%
30	1.018	0.889	0.665	34.62%	25.16%
35	1.359	1.173	0.864	36.45%	26.36%
40	1.753	1.493	1.087	38.01%	27.24%
45	2.194	1.851	1.333	39.23%	27.98%
50	2.687	2.250	1.609	40.11%	28.47%

Table 8. Average CPU elapsed time (sec) versus the maximum order of moments p_{max} for the computation of ZMs for the proposed method, method of [37] and Algorithm E of [34] on 12 different images of size 512×512

Maximum order of ZMs (p_{max})	**Method of [37] (A)**	**Algorithm E of [34] (B)**	**Proposed method (C)**	**% improvement $\frac{(A-C)}{A} \times 100$**	**% improvement $\frac{(B-C)}{B} \times 100$**
5	0.203	0.195	0.161	20.81%	17.56%
10	0.608	0.576	0.457	24.86%	20.61%
15	1.217	1.137	0.874	28.19%	23.12%
20	2.045	1.884	1.410	31.06%	25.18%
25	3.076	2.789	2.042	33.60%	26.77%
30	4.326	3.857	2.775	35.85%	28.04%
35	5.780	5.098	3.602	37.68%	29.34%
40	7.453	6.507	4.529	39.24%	30.41%
45	9.329	8.077	5.555	40.46%	31.23%
50	11.424	9.835	6.702	41.34%	31.86%

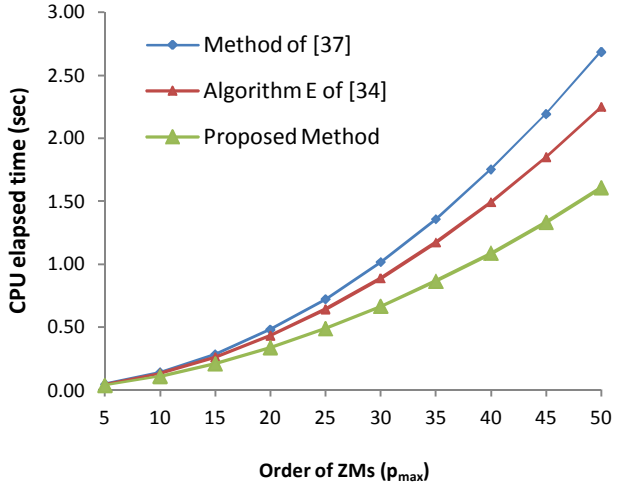


Fig. 4. Average CPU elapsed time (sec) versus the maximum order of moments p_{max} for the computation of ZMs for the proposed method, method of [37] and Algorithm E of [34] on 12 different images of size 256×256.

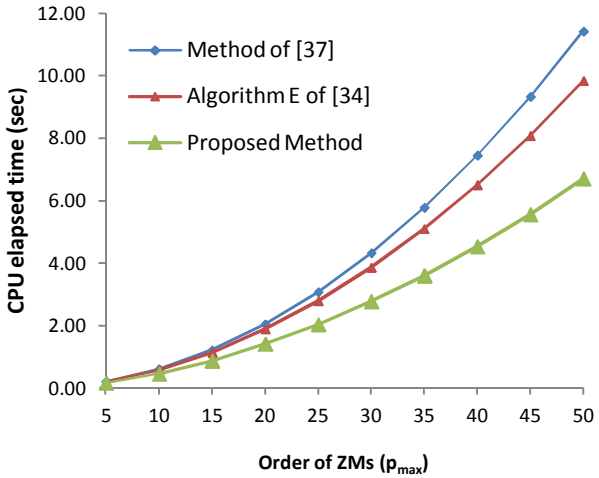


Fig. 5 Average CPU elapsed time (sec) versus the maximum order of moments p_{max} for the computation of ZMs for the proposed method, method of [37] and Algorithm E of [34] on 12 different images of size 512×512.

5.2. Fast Computation of ZMs Using The Proposed Convolution Model on Single Image

In this set of experiments, we obtain the computation time to compute ZMs over blocks of size $n \times n$ centered at each pixel location in the given image. So, if the image is of size $M \times N$ pixels, then the ZMs are computed for MN number of blocks, each of size $n \times n$. Here also, the ZMs kernel functions are precomputed only once for a block and stored in a table. In our experimental set up, a standard gray scale *Lena* image of size 512×512 pixels has been considered to compute ZMs over blocks of size 7×7 (i.e., $n = 7$). The value of p_{max} is varied from order 5 to order 15 with equal increments of one. Here, the values of the size of blocks and number of moments are taken in accordance with their values used in image denoising and image super-resolution. The experiment is repeated for $n = 9$ and 11, i.e, we compute ZMs with blocks of size 9×9 and 11×11, respectively. Tables 9-11 present the computation time versus the maximum number of moments $p_{max} (p_{max} \in \{5,6, \ldots, 15\})$ for blocks of size 7×7, 9×9 and 11×11, respectively, using the convolution process of [37], Algorithm E of [34] and the proposed convolution model. From the tables it can be observed that the proposed approach yields a maximum improvement in the computation speed of 31.70%, 32.97% and 33.17% over the convolution process of [37] and a maximum improvement of 26.88%, 28.24% and 28.47% over Algorithm E of [34], for the blocks of size 7×7, 9×9 and 11×11, respectively. It is also observed here that the higher is the block size, the better is the improvement in speed. This is due to the fact that as the block size increases, the number of convolutions required to compute ZMs will increase. Thus, the proposed convolution model reduces more number of operations, resulting in better speed improvements over the compared methods.

Table 9. CPU elapsed time (sec) versus the maximum order of moments p_{max} for the computation of ZMs over block of size 7×7 on *Lena* image of size 512×512, for the proposed method, method of [37] and Algorithm E of [34]

Maximum order of ZMs (p_{max})	**Method of [37] (A)**	**Algorithm E of [34] (B)**	**Proposed method (C)**	**% improvement $\frac{(A-C)}{A} \times 100$**	**% improvement $\frac{(B-C)}{B} \times 100$**
5	0.325	0.312	0.279	14.10%	10.58%
6	0.407	0.390	0.347	14.79%	11.03%
7	0.507	0.484	0.424	16.33%	12.40%
8	0.589	0.561	0.484	17.82%	13.73%
9	0.706	0.671	0.566	19.87%	15.65%
10	0.808	0.765	0.625	22.67%	18.30%
11	0.948	0.895	0.718	24.26%	19.78%
12	1.077	1.014	0.801	25.61%	21.01%
13	1.229	1.154	0.889	27.64%	22.96%
14	1.404	1.315	0.998	28.89%	24.11%
15	1.621	1.514	1.107	31.70%	26.88%

Table 10. CPU elapsed time (sec) versus the maximum order of moments p_{max} for the computation of ZMs over block of size 9×9 on *Lena* image of size 512×512, for the proposed method, method of [37] and Algorithm E of [34]

Maximum order of ZMs (p_{max})	**Method of [37] (A)**	**Algorithm E of [34] (B)**	**Proposed method (C)**	**% improvement $\frac{(A-C)}{A} \times 100$**	**% improvement $\frac{(B-C)}{B} \times 100$**
5	0.572	0.549	0.484	15.31%	11.84%
6	0.713	0.683	0.593	16.85%	13.18%
7	0.860	0.821	0.702	18.34%	14.49%
8	1.014	0.966	0.811	20.03%	16.05%
9	1.197	1.137	0.936	21.80%	17.68%
10	1.439	1.362	1.092	24.11%	19.82%
11	1.652	1.560	1.232	25.44%	21.03%
12	1.855	1.747	1.357	26.85%	22.32%
13	2.143	2.013	1.513	29.40%	24.84%
14	2.461	2.306	1.700	30.93%	26.28%
15	2.839	2.652	1.903	32.97%	28.24%

Table 11. CPU elapsed time (sec) versus the maximum order of moments p_{max} for the computation of ZMs over block of size 11×11 on *Lena* image of size 512×512, for the proposed method, method of [37] and Algorithm E of [34]

Maximum order of ZMs (p_{max})	**Method of [37] (A)**	**Algorithm E of [34] (B)**	**Proposed method (C)**	**% improvement $\frac{(A-C)}{A} \times 100$**	**% improvement $\frac{(B-C)}{B} \times 100$**
5	0.861	0.827	0.718	16.60%	13.18%
6	1.108	1.061	0.905	18.31%	14.70%
7	1.318	1.259	1.061	19.51%	15.73%
8	1.562	1.488	1.232	21.14%	17.20%
9	1.815	1.724	1.404	22.64%	18.56%
10	2.160	2.044	1.638	24.15%	19.86%
11	2.528	2.387	1.872	25.95%	21.58%
12	2.849	2.683	2.091	26.61%	22.06%
13	3.256	3.058	2.309	29.08%	24.49%
14	3.776	3.538	2.605	31.02%	26.37%
15	4.342	4.056	2.902	33.17%	28.45%

6. Conclusion

A new convolution model for the fast computation of the ZMs is presented in this paper. The computation of ZMs consists of two parts: the computation of the kernel function and the convolution of the image data with the kernel function. Already, fast algorithms exist for the computation of the kernel functions of the ZMs. In this paper, an attempt has been made for the fast convolution of the image data with the kernel function which reduces the number of additions/subtractions from 56 to 24 and the number of multiplications from 12 to 8 at each location of one octant of the circular disk on which the moments are computed. The experimental results show that the speed of the computation of the moments increases by a factor varying from 15% to 41% for the computation of ZMs for multiple images as compared to the existing fastest algorithms available in the literature. When ZMs are computed at each pixel of an image on overlapping blocks, the improvement in computation time varies from 10% to 33%. It is also observed that the higher is the moment order, the better is the enhancement in the speed. The accuracy of the moments, their immunity to noise and image reconstruction ability is not affected by the proposed fast convolution model. Their values are obtained as accurately as provided by the existing recursive and fast methods.

Acknowledgements

The comments and suggestions provided by the anonymous reviewers to raise the standard of the paper are highly appreciated. The authors are grateful to the University Grants Commission (UGC), New Delhi, India, for providing financial grants for the Major Research Project entitled, “Development of Efficient Techniques for Feature Extraction and Classification for Invariant Pattern Matching and Computer Vision Applications”, vide its File No.: 43-275/2014(SR). The financial grant provided by the UGC, New Delhi, India, to one of the authors (Ashutosh Aggarwal) under the SAP-III programme is also highly acknowledged

References

[1] M.R. Teague, Image analysis via the general theory of moments, J. Opt. Soc. Am. 70 (1980) 920–930.

[2] A. Broumandnia, J. Shanbehzadeh, Fast Zernike wavelet moments for Farsi character recognition, Image Vis. Comput. 25 (2007) 717–729.

[3] C. Kan, M.D. Srinath, Invariant character recognition with Zernike and orthogonal Fourier–Mellin moments, Pattern Recogn. 35 (2002) 143–154.

[4] A. Khotanzad, J.-H. Lu, Classification of invariant image representations using a neural network, IEEE Trans. Acoust., Speech, Signal Process. 38 (1990) 1028–1038.

[5] L. Wang, G. Healey, Using Zernike moments for the illumination and geometry invariant classification of multispectral texture, IEEE Trans. Image Process. 7 (1998) 196–203.

[6] G.A. Papakostas, Y.S. Boutalis, D.A. Karras, B.G. Mertzios, A new class of Zernike moments for computer vision applications, Inform. Sci. 177 (2007) 2802–2819.

[7] P. Raveendran, S. Omatu, Performance of an optimal subset of Zernike features for pattern classification, Inform. Sci. 1 (1994) 133–147.

[8] C.-Y. Wee, R. Paramesran, F. Takeda, New computational methods for full and subset Zernike moments, Inform. Sci. 159 (2004) 203–220.

[9] Z. Chen, S.-K. Sun, A Zernike moment phase descriptor for local image representation and matching, IEEE Trans. Image Process. 19 (2010) 205–219.

[10] Y.-S. Kim, W.-Y. Kim, Content-based trademark retrieval system using visually salient features, in: Proceedings of 1997 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 1997, pp. 307–312.

[11] S. Li, M.-C. Lee, C.-M. Pun, Complex Zernike moments features for shape based image retrieval, IEEE Trans. Syst., Man, Cybern – Part A: Syst. Hum. 39 (2009) 227–237.

[12] C. Singh, Pooja, Improving image retrieval using combined features of Hough transform and Zernike moments, Opt. Lasers Eng. 49 (2011) 1384–1396.

[13] T.F. Ansary, M. Daoudi, J.-P. Vandeborre, A Bayesian 3D search engine using adaptive views clustering, IEEE Trans. Multimedia 9 (2007) 78–88.

[14] S. Hou, K. Ramani, Calligraphic interfaces: classifier combination for sketch-based 3D part retrieval, Comput. Graph. 31 (2007) 598–609.

[15] N. Bissantz, H. Holzmann, M. Pawlak, Testing for image symmetries-with application to confocal microscopy, IEEE Trans. Inform. Theory 55 (2009) 1841–1855.

[16] W-Y. Kim, Y-S Kim, Robust rotation angle estimator, IEEE Trans. Pattern Anal. Mach. Intell. 21 (1999) 768–773.

[17] J. Revaud, G. Lavoue, A. Baskurt, Improving Zernike moments comparison for optimal similarity and rotation angle retrieval, IEEE Trans. Pattern Anal. Mach. Intell. 31 (2009) 27–636.

[18] Z. Ji, Q. Chen, Q.-S. Sun, D.-S. Xia, A moment-based nonlocal-means algorithm for image denoising, Information Processing Letters 109 (2009) 1238–1244.

[19] X. Gao, Q. Wang, X. Li, D. Tao, K. Zhang, Zernike moment-based image super resolution, IEEE Trans. Image Proc. 20 (2011) 2738–2747.

[20] R. Gayathri, P. Ramamoorthy, Automatic Palmprint Identification based on High Order Zernike Moment, American Journal of Applied Sciences 9 (2012) 759-765.

[21] J. Haddadnia, M. Ahmadi, K. Faez, An efficient method for recognition of human faces using higher orders Pseudo Zernike Moment Invariant, in: Proceedings of Fifth IEEE International Conference on Automatic Face and Gesture Recognition, Washington, DC, USA, 2002, pp. 330 – 335

[22] G. Amayeh, G. Bebis, A. Erol, M. Nicolescu, Peg-free hand shape verification using high order Zernike moments, in: Proceedings of 2006 Conference on Computer Vision and Pattern Recognition Workshop (CVPRW'06), 2006, pp. 40-40.

[23] H.S. Kim, H.K. Lee, Invariant image watermarking using Zernike moments, IEEE Trans. Image Process. 13 (2003) 766–775.

[24] C. Singh, S.K. Ranade, A high capacity image adaptive watermarking scheme with radial harmonic Fourier moments. Digital Signal Processing, Digital Signal Processing 23 (2013) 1470-1482

[25] C. Singh, S.K. Ranade, Image adaptive and high-capacity watermarking system using accurate Zernike moments, IET Image Processing 8 (2014) 373-382.

[26] M. Pawlak, On the reconstruction aspect of moment descriptors, IEEE Trans. Inform. Theory 38 (1992) 1698–1708.

[27] C. Singh, Improved quality of reconstructed images using floating point arithmetic for moment calculation, Pattern Recogn. 39 (2006) 2047–2064.

[28] D. Hong, S. Liao, Central-symmetrical property analysis on circularly orthogonal moments, Journal of Theoretical and Computer Science 8 (2014) 11-26.

[29] A.W. Deng, C.H. Wei, C.Y. Gwo, Stable, fast computation of high-order Zernike moments using a recursive method, Pattern Recognition 56 (2016) 16-25.

[30] E.C. Kitner, A recursive relation for calculating the Zernike polynomials, Opt. Acta 23 (1976) 499-500.

[31] A. Prata, W.V.T. Rusch, Algorithm for computation of Zernike polynomials expansion coefficients, Appl. Opt. 28 (1989) 749–754.

[32] C.-W. Chong, R. Paramesran, R. Mukundan, A comparative analysis of algorithms for fast computation of Zernike moments, Pattern Recogn. 36 (2003) 731–742.

[33] C. Singh, E. Walia, Fast and numerically stable methods for the computation of Zernike moments, Pattern Recogn. 43 (2010) 2497–2506.

[34] C. Singh, E. Walia, Algorithms for fast computation of Zernike moments and their numerical stability, Image Vis. Comput. 29 (2011) 251–259.

[35] C. Singh, E. Walia, Pooja, R. Upneja, Analysis of algorithms for fast computation of pseudo Zernike moments and their numerical stability, Digital Signal Processing 22 (2012) 1031-1043.

[36] S.-K. Hwang, W.-Y. Kim, A novel approach to the fast computation of Zernike moments, Pattern Recog. 39 (2006) 2065–2076

[37] C.-Y. Wee, R. Paramesaran, Efficient computation of radial moment functions using symmetrical property, Pattern Recog. 39 (2006) 2036–2046.

[38] R. Upneja, C. Singh, Fast computation of Jacobi-Fourier moments for invariant image recognition, Pattern Recogn. 48(2015)1836–1843.

[39] C. Loeffler, A. Ligtenberg, G.S. Moschytz, Practical fast 1-D DCT algorithms with 11 multiplications, in: Proceedings of IEEE Int. Conf. Acoustics, Speech and Signal Processing, vol. 2, May 1989, pp. 988–991.

[40] H. Zhu, Y. Yang, X. Zhu, Z. Gui, H. Shu, General form for obtaining unit disc-based generalized orthogonal moments, IEEE Trans. Image Process. 23 (2014) 5455-5469.

[41] C.Y. Wee, R. Paramesaran, On the computational aspects of Zernike moments, Image Vis. Comput. 25 (2007) 967–980.

[42] C. Singh, E. Walia, R. Upneja, Accurate calculation of Zernike moments, Inform. Sci. 233 (2013) 255-275.

[43] C. Singh, A. Kaur, Fast computation of polar harmonic transforms, J. Real-Time Image Proc. 10 (2015) 59-66.