J Real-Time Image Proc (2015) 10:59–66
DOI 10.1007/s11554-012-0252-y

ORIGINAL RESEARCH PAPER

Fast computation of polar harmonic transforms

Chandan Singh · Amandeep Kaur

Received: 9 November 2011 / Accepted: 17 April 2012 / Published online: 17 May 2012
© Springer-Verlag 2012

Abstract Polar harmonic transforms (PHTs) are orthogonal rotation invariant transforms that provide many numerically stable features. The kernel functions of PHTs consist of sinusoidal functions that are inherently computation intensive. We develop a fast approach for their computation using recursion and 8-way symmetry/anti-symmetry property of the kernel functions. The clustering of pixels at eight radially symmetrical locations enhances the speed of computation. Experimental results show that the proposed method is faster by a factor lying between three and four compared to the existing fast method.

Keywords Orthogonal transforms · Orthogonal moments · PHTs · Rotation invariance

1 Introduction

Region-based image descriptors capture the global properties of the pixel distribution in the entire image, and hence they are extensively used in many pattern recognition and computer vision applications. Rotation invariant pattern representation is an essential requirement because images that are rotated versions of each other perceived to be the same by human beings. Therefore, the pattern recognition systems should mimic the human perception by presenting methods that handle rotation, change in size and location. Rotation invariant moments and transforms are such processes, which successfully deal with these situations. There are two types of rotation invariant moments and transforms: orthogonal and non orthogonal. The orthogonal rotation invariant moments (ORIMs) and orthogonal rotation invariant transforms (ORITs) are more effective in performance because they have minimum information redundancy and hence better information compactness. A few low order moments and transforms are sufficient to capture the essential features of an image. Among the ORIMs Zernike moments (ZMs), pseudo Zernike moments (PZMs) and orthogonal Fourier Mellin moments (OFMMs) [1, 2] are the most popular. The ORITs that were introduced recently by Yap et al. [3] include the polar complex exponential transforms (PCET), polar cosine transforms (PCT) and polar sine transforms (PST). These transforms are collectively known as PHTs. The difference between ORIMs and ORITs is that the radial parts of the kernel functions in ORIMs are polynomials and in ORITs these are sinusoidal functions. The PHTs are preferred to ORIMs because PHTs are computationally very fast [4] and the high order transforms are numerically stable, where as the ORIMs are less efficient and high order moments are numerically unstable. Because of their attractive features, PHTs have recently been used in many image processing applications. Liu et al. [5] observed that PHTs-based features yield results comparable to state-of-the-art methods for fingerprint classification. An extensive evaluation of invariance property of PHTs for image representation in terms of rotation, scale and noise has been conducted by Li et al. [6]. The authors observed that the ORITs are more suitable than ORIMs for applications, which require many features. The results are compared with ZMs and PZMs, and it is observed that the performance of PHTs is better than that of ZMs and PZMs. Recently, Miao, et al. [7] have applied PHTs on Radon

C. Singh (✉) · A. Kaur
Department of Computer Science,
Punjabi University Patiala, Punjab, India
e-mail: chandan.csp@gmail.com

A. Kaur
e-mail: aman_k2007@hotmail.com

images for object recognition. They have also compared their results with that obtained by ZMs, OFFMs and Radial Fourier Mellin moments. Through the theoretical analysis and experimental results, it is observed that the performance of PHTs for image description is much better than the three moments especially under noisy conditions. Hoang and Tabbone [8] introduce a class of harmonic radial kernels that satisfy orthogonality conditions and derive PCET as a special case. Experimental results prove the effectiveness of this class of transforms for the description performance and pattern recognition ability. Singh et al. [9] perform the image reconstruction analysis of PHTs and show that PCET provide much better reconstructed images than ORIMs and other types of PHTs.

In this paper, we propose a method for the fast computation of the PHTs by developing recursive relations for the radial and angular parts of the kernel functions of the transform. An 8-way symmetry/anti-symmetry property is used to enhance the speed of the algorithm. The clustering of pixels at eight radially symmetric location's results in a fast algorithm. These algorithms can be used for invariant pattern and shape recognition problems in real-time environment or on devices with low computation power.

The rest of the paper is organized as follows. An overview of PHTs with their discrete computational framework is presented in Sect. 2. The fast recursive relations, 8-way symmetry/anti-symmetry, and clustering of image pixels are discussed in Sect. 3. The speed performance experimental results are given in Sect. 4. Conclusion is discussed in Sect. 5.

2 Polar harmonic transforms

Polar harmonic transforms consist of the polar complex exponential transform (PCET), PCT and PST [3]. They have identical mathematical representation with a difference in the radial part of the kernel function. Let $f(r,\theta)$ be a continuous image function defined on a unit disk $D = \{(r,\theta)\, 0 \le r \le 1,\ 0 \le \theta \le 2\pi\}$. The PHTs of order n and repetition m are defined by

$$A_{nm} = \lambda \int_0^{2\pi} \int_0^1 f(r,\theta)\, V_{nm}^*(r,\theta)\, r\mathrm{d}r\mathrm{d}\theta \tag{1}$$

where $n,\ m = 0, \pm 1, \pm 2,\ldots$. The kernel function $V_{nm}^*(r,\theta)$ is the complex conjugate of the function $V_{nm}(r,\theta)$ determined by

$$V_{nm}(r,\theta) = R_n(r) e^{jm\theta} \tag{2}$$

with $j = \sqrt{-1}$. The radial part of the kernel function and the parameter λ are expressed as

$$\text{PCET} : R_n(r) = e^{j2\pi n r^2},\ \lambda = \frac{1}{\pi} \tag{3}$$

$$\text{PCT and PST} : R_n(r) = \begin{cases} \cos(\pi n r^2), & \text{for PCT} \\ \sin(\pi n r^2), & \text{for PST} \end{cases} \tag{4}$$

$$\lambda = \begin{cases} \frac{1}{\pi}, & n = 0 \\ \frac{2}{\pi}, & n \ne 0 \end{cases} \tag{5}$$

The radial part of the kernel function satisfies the orthogonality condition

$$\int_0^1 R_n(r)[R_{n'}(r)]^* r dr = \frac{1}{2}\delta_{nn'} \tag{6}$$

where $\delta_{nn'} = 1$ if $n = n'$, and 0 otherwise.

Also, the complete kernel function $V_{nm}(r,\theta)$ satisfies the orthogonality condition

$$\int_0^{2\pi} \int_0^1 V_{nm}(r,\theta)[V_{n'm'}(r,\theta)]^* r\mathrm{d}r\mathrm{d}\theta = \pi \delta_{nn'}\delta_{mm'} \tag{7}$$

The orthogonality property of the kernel function enables the image function $f(r,\theta)$ to be reconstructed

$$\hat{f}(r,\theta) = \sum_{n=-n_{\max}}^{n_{\max}} \sum_{m=-m_{\max}}^{m_{\max}} A_{nm} V_{nm}(r,\theta) \tag{8}$$

where $n_{\max}$ and $m_{\max}$ are the highest allowed values of n and m, respectively. Greater number of transform coefficients used in the reconstruction helps $\hat{f}(r,\theta)$ to be closer to $f(r,\theta)$.

The magnitude of transform coefficients is rotation invariant. Let A_{nm}^{α} be the transform coefficients of the image function $f(r,\ \theta + \alpha)$ obtained by rotating an image by an angle α about its center, then it can be shown that

$$A_{nm}^{\alpha} = A_{nm} e^{-jm\alpha} \tag{9}$$

Thus, yielding a very important property of rotation invariance of its magnitude, namely, $|A_{nm}^{\alpha}| = |A_{nm}|$.

Deriving PHTs using Eq. (1) is difficult because in digital image processing, the image function $f(r,\theta)$ is discrete, and defined in rectangular coordinates system. Let $N \times N$ be an image with (i, k) representing a pixel at the ith row and jth column. We perform a mapping of pixel's grid from $N \times N$ square domain to $[-1,1] \times [-1,1]$ with the help of the following transformations

$$x_i = \frac{2i + 1 - N}{N}, y_k = \frac{2k + 1 - N}{N}, i,k = 0,1,\ldots,N-1 \tag{10}$$

with $\Delta x = \Delta y = \frac{2}{N}$

The unit continuous disk $D = \{f(r,\theta)\, 0 \le r \le 1, 0 \le \theta \le 2\pi\}$ is now approximated by a unit digital disk $D' = \{f(i,k),\, x_i^2 + y_k^2 \le 1\}$. The mapping process allows only those pixels to belong to D' whose centers $(x_i,\, y_k)$ lie within the unit disk D'. This approximation is depicted in Fig. 1a and b, where Fig. 1a is an 8 × 8 pixels grid and the continuous unit disk D and Fig. 1b represents the approximated unit disk D′ and the pixels whose centers lie within the unit disk D′ are shaded with gray color. Thus, Eq. (1) for the computation of PHTs assume the form

$$A_{nm} = \lambda \int\limits_{\substack{x=-1 \\ x_i^2 + y_k^2 \le 1}}^{1} \int\limits_{y=-1}^{1} f(x,y) V_{nm}^*(x,y) \mathrm{d}x\mathrm{d}y \tag{11}$$

where $r = \sqrt{x^2 + y^2}$ and $\theta = \tan^{-1}(y/x)$, and $V_{nm}^*(x,y)$ derives from Eq. (2) with the change of coordinates from polar system to Cartesian system.

The image function $f(x,y)$ is discrete. Assuming that the function value $f(x,y) = f(i,k)$ is constant over a pixel grid $\left[x_i - \frac{\Delta x}{2},\, y_k - \frac{\Delta y}{2}\right] \times \left[x_i + \frac{\Delta x}{2}, y_k + \frac{\Delta y}{2}\right]$, then Eq. (11) can be approximated by

$$A_{nm} = \lambda \sum_{i=0}^{N-1} \sum_{\substack{k=0 \\ x_i^2 + y_k^2 \le 1}}^{N-1} f(i,k) \int\limits_{x_i - \frac{\Delta x}{2}}^{x_i + \frac{\Delta x}{2}} \int\limits_{y_k - \frac{\Delta y}{2}}^{y_k + \frac{\Delta y}{2}} V_{nm}^*(x,y) \mathrm{d}x\mathrm{d}y \tag{12}$$

No analytical solution exists for double integration involved in the right hand side of Eq. (12), therefore, normally a zeroth order approximation is used [3].

$$A_{nm} = \frac{4\lambda}{N^2} \sum_{i=0}^{N-1} \sum_{\substack{k=0 \\ x_i^2 + y_k^2 \le 1}}^{N-1} f(i,k) V_{nm}^*(x_i, y_k) \tag{13}$$

Alternatively, PHTs can be computed in polar coordinates as suggested by [10, 11] for ORIMs such as the ZMs, PZMs and OFMMs. However, like ORIMs, computation of PHTs in polar coordinates requires reconfiguration of image pixels in polar coordinates. This process involves interpolation error because the pixel intensities are required to be determined at new locations [10, 11]. Therefore, normally PHTs are computed using Eq. (13) as explained in [3].

3 Fast computations of kernel functions

An attractive characteristic of PHTs is their low computation requirement of kernel functions as compared to ORIMs whose kernel functions are computation intensive because of the presence of high order of polynomials and factorial terms. Nevertheless, the kernel functions of PHTs comprise the sinusoidal functions. The computation cost of these functions is still high. In a recent paper by Yang et al. [4], 8-way symmetry/anti-symmetry is used for the calculation of the radial and angular sinusoidal functions, which increases the computation speed by a factor of six to eight. The time taken for the evaluation of trigonometric functions is very high. Thus, the computation of sinusoidal functions even in an octant of an image is time consuming. The high time requirement is gauged from the fact that the evaluation of cosine and sine functions of an image of size $N \times N$ pixels (with $N =$ even) is required to be computed at $N(N+2)/8$ locations over an octant and if the maximum order and repetition are $n_{\max}$ and $m_{\max}$, respectively, then the time complexity is $O(N^2 n_{\max} m_{\max})$ which is very high. This time complexity cannot be reduced even after saving these values in tables. We propose recursive algorithms for the calculation of trigonometric functions

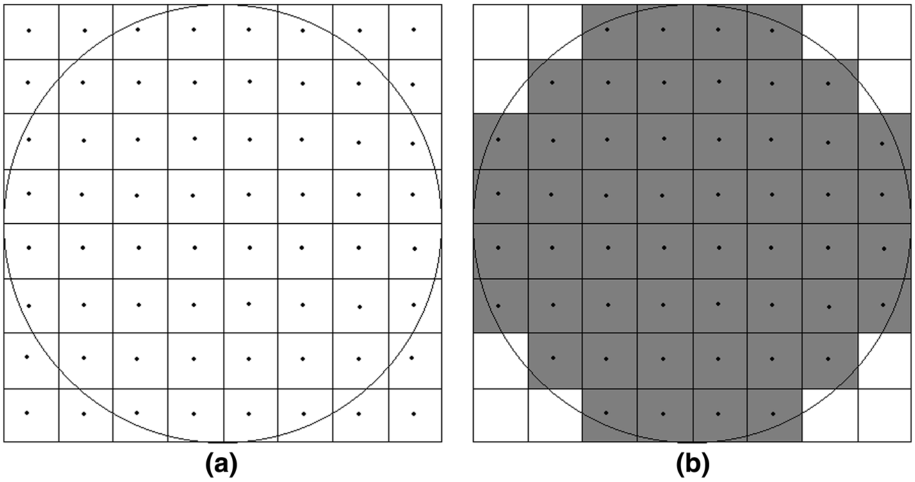

Fig. 1 **a** An 8 × 8 pixel grid and the *continuous inscribed circle* D, and **b** the approximation of the *circle* D′

for the radial and angular parts of the kernel functions. It is shown that while the angular part does not involve the computation of trigonometric functions through inbuilt library functions even for once, the radial part requires their evaluation only once at a pixel location and the higher order trigonometric functions are computed recursively. The proposed method is similar to the recursive approach adopted for angular functions for the calculation of ZMs by Singh and Walia [12].

3.1 Computation of angular functions

The angular functions $e^{jm\theta} = \cos(m\theta) + j\sin(m\theta)$ can be computed efficiently for $m = 0, 1, 2, \ldots, m_{\max}$ at a pixel location (*i*, *k*). For this purpose, we use the following recurrence relations to evaluate $\cos(m\theta)$ and $\sin(m\theta)$.

$$\cos((m+1)\theta) = \cos(\theta)\,\cos(m\theta) - \sin(\theta)\,\sin(m\theta) \qquad (14)$$

$$\sin((m+1)\theta) = \sin(\theta)\,\cos(m\theta) + \cos(\theta)\,\sin(m\theta) \qquad (15)$$

where $m = 0, 1, 2, \ldots, m_{\max}$, $r = \sqrt{x_i^2 + y_k^2}$, $\cos(\theta) = x_i/r$, $\sin(\theta) = y_k/r$, x_i and y_k are given in Eq. (10), and $\cos(0) = 1.0$ and $\sin(0) = 0.0$. We use two tables, say, cos_*theta*$[m_{\max} + 1]$ and sin_*theta*$[m_{\max} + 1]$, to save the values in memory. The extra space required for saving the tables is $2(m_{\max} + 1)$ words. We compute $\cos(m\theta)$ directly using library function and through recursion for $m = 15$ and 40, and $\theta = \pi/8, \pi/7, \pi/6, \pi/3,$ and $\pi/2$. The computation is performed in Microsoft's Visual C ++6.0 under Windows environment with double precision arithmetic, which provides accuracy up to 15 decimal places. The results are quoted in Table 1 and show that the error accumulated during recursion is negligible. The differences in results are very small, which are of the order between 10^{-15} and 10^{-16} and consistent with the finite precision arithmetic being used.

3.2 Computation of radial functions

We now focus toward the computation of radial function of PCET. The calculation for PCT and PST follows the similar steps. The radial functions $e^{j2\pi nr^2} = \cos(2\pi nr^2) + j\,\sin(2\pi nr^2)$ can be computed recursively for different orders n. For the order $n = 1$, the computation of $\cos(2\pi r^2)$ and $\sin(2\pi r^2)$ is performed directly using the library functions. For orders $n > 1$, Eqs. (16) and (17) are used to compute $\cos(2\pi nr^2)$ and $\sin(2\pi nr^2)$ recursively.

$$\cos(2\pi r^2(n+1)) = \cos(2\pi r^2)\,\cos(2\pi r^2 n) - \sin(2\pi r^2)\,\sin(2\pi r^2 n) \qquad (16)$$

$$\sin(2\pi r^2(n+1)) = \sin(2\pi r^2)\,\cos(2\pi r^2 n) + \cos(2\pi r^2)\,\sin(2\pi r^2 n) \qquad (17)$$

where $n = 1, 2, 3, \ldots, n_{\max}$

Table 1 Comparison of values of $\cos(m\theta)$ computed directly using library function and using recursion, Eqs. (14) and (15)

m/θ	15			40		
	$\cos(m\theta)$			$\cos(m\theta)$		
	Direct (*A*)	Recursion (*B*)	(*A*–*B*)	Direct (*C*)	Recursion (*D*)	(*C*–*D*)
$\pi/8$	9.238795325112865e−001	9.238795325112867e−001	− 2.401507812055392e−016	− 1.000000000000000e+000	− 9.999999999999991e−001	− 8.881784197001252e−016
$\pi/7$	9.009688679024196e−001	9.009688679024195e−001	3.539920093165172e−017	6.234898018587329e−001	6.234898018587335e−001	− 6.109479241955817e−016
$\pi/6$	1.194330008155720e−015	1.332267629550187e−015	− 1.379376213944683e−016	− 5.000000000000003e−001	− 4.999999999999970e−001	− 3.316140764764163e−015
$\pi/3$	− 1.000000000000000e+000	− 1.000000000000000e+000	0.000000000000000e+000	− 4.999999999999994e−001	− 5.000000000000046e−001	5.300041005032452e−015
$\pi/2$	− 2.694811604767033e−015	− 9.184547653667829e−016	− 1.776356839400250e−015	1.000000000000000e+000	1.000000000000000e+000	0.000000000000000e+000

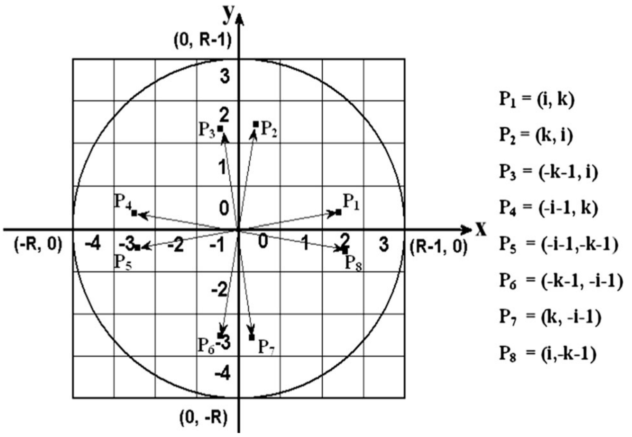

Fig. 2 8-way symmetry of a pixel $P_1(i,k)$ with $x_i = (2i+1)/N$, $y_k = (2k+1)/N$. The origin of the coordinate system is at the center of the image

3.3 Eight way symmetry/anti-symmetry of radial and angular functions

The radial functions $e^{j2\pi nr^2}$ of PCET assume the same values at 8-radially symmetric locations denoted by P_1 through P_8 within a circle as shown in Fig. 2. Let the center of the circle be at origin (0,0) with the digital radius $R = N/2$. Let a pixel $P_1(i,k)$, $i = 0,1,2,\ldots,R-1$; $k = 0,1,2,\ldots,i$ be given in the first octant of the circle with the coordinates of its center (the center of (i,=k) pixel grid) $x_i = (2i+1)/N$, $y_k = (2k+1)/N$, mapped within the unit disk, then $r = \sqrt{x_i^2 + y_k^2}$. The seven other pixels located at radially symmetric locations P_2 through P_8 are given by$P_2(k,i)$, $P_3(-k-1,\, i)$, $P_4(-i-1,k)$, $P_5\,(-i-1,\, -k-1)$, $P_6(-k-1, -i-1)$, $P_7(k,-i-1)$, and $P_8(i,-k-1)$. It can be easily verified that the centers of these pixel grids have the same values of r. Since we are referring the pixels w.r.t. the center of the digital image assumed to be at (0,0), we have negative indices while referring them. In actual implementation, we add R with each pixel location. For example, the pixel $P_1(i,k)$ will be referred to as $P_1(R+i, R+k)$, $i = 0,1,2,\ldots,R-1$, $k = 0,1,2,\ldots,i$, meaning thereby that the origin of the image is at the top left corner of the square domain. When $k = i$, 4-way symmetry is considered instead of 8-way symmetry because of the digital nature of the circle. Moreover, in the above discussion, we have assumed N to be even. However, the treatment is the same for odd N except for a pixel at the center of the image for which the computation is performed separately. Also for odd N, 4-way symmetry exists for $k = 0$ (along axes) and $k = i$ (along diagonals).

Further, if the pixel (i, k) makes an angle θ with x-axis with $\theta = \tan^{-1}(y_k/x_i)$, than the seven radially symmetric pixels P_2 through P_8 will make angles $\frac{\pi}{2} - \theta$, $\frac{\pi}{2} + \theta$, $\pi - \theta$, $\pi + \theta$, $\frac{3\pi}{2} - \theta$, $\frac{3\pi}{2} + \theta$, and $2\pi - \theta$, respectively. Therefore, if $\cos(m\theta)$ and $\sin(m\theta)$ are evaluated at P_1, then their values can be used for locations P_2 through P_8 as shown in Tables 2 and 3. This situation is further explained in Sect. 3.4. The symmetry/anti-symmetry property reduces the requirement of performing the calculations from N^2 locations to $N(N+2)/8$ locations.

3.4 Clustering of image pixels

Significant enhancement in speed can be achieved by clustering the eight symmetrically located pixels. The clustering of pixels is achieved using 8-way symmetry/anti-symmetry property of radial and angular parts of kernel functions, explained by rewriting Eq. (13) in the form

$$A_{nm} = \frac{4\lambda}{N^2} \sum_{i=0}^{N-1} \sum_{\substack{k=0 \\ x_i^2 + y_k^2 \le 1}}^{N-1} f(i,k) \times \begin{bmatrix} \{\cos(2\pi n r_{ik}^2)\cos(m\theta_{ik}) - \sin(2\pi n r_{ik}^2)\sin(m\theta_{ik})\} - \\ j\{\sin(2\pi n r_{ik}^2)\cos(m\theta_{ik}) + \cos(2\pi n r_{ik}^2)\sin(m\theta_{ik})\} \end{bmatrix} \tag{18}$$

where $r_{ik}^2 = \sqrt{x_i^2 + y_k^2}$, $\theta_{ik} = \tan^{-1}(y_k/x_i)$. The 8-way symmetry/anti-symmetry property leads Eq. (18) to

$$A_{nm} = \frac{4\lambda}{N^2} \sum_{i=0}^{R-1} \sum_{k=0}^{i} \begin{bmatrix} \left\{ \left(\sum_{u=1}^{8} f_u \cos(\theta_u) \right) \cos(2\pi n r_{ik}^2) - \left(\sum_{u=1}^{8} f_u \sin(\theta_u) \right) \sin(2\pi n r_{ik}^2) \right\} - \\ j\left\{ \left(\sum_{u=1}^{8} f_u \cos(\theta_u) \right) \sin(2\pi n r_{ik}^2) + \left(\sum_{u=1}^{8} f_u \sin(\theta)_u \right) \cos(2\pi n r_{ik}^2) \right\} \end{bmatrix} \tag{19}$$

where $f_1 = f(R+i, R+k), f_2 = f(R+k, R+i), \ldots, f_8 = f(R+i, R-k-1), \theta_1 = m\theta_{ik}, \theta_2 = m(\frac{\pi}{2} - \theta_{ik}), \ldots, \theta_8 = m(2\pi - \theta_{ik}),$

$$A_{nm} = \frac{4\lambda}{N^2} \sum_{i=0}^{R-1} \sum_{k=0}^{i} \begin{bmatrix} [\{F_1\cos(m\theta_{ik}) + F_2\sin(m\theta_{ik})\}\cos(2\pi n r_{ik}^2) - \{F_3\cos(m\theta_{ik}) + F_4\sin(m\theta_{ik})\}\sin(2\pi n r_{ik}^2)] - \\ j[\{F_1\cos(m\theta_{ik}) + F_2\sin(m\theta_{ik})\}\sin(2\pi r_{ik}^2) + \{F_3\cos(m\theta_{ik}) + F_4\sin(m\theta_{ik})\}\cos(2\pi n r_{ik}^2)] \end{bmatrix} \tag{20}$$

Table 2 Values of $\cos(m\theta)$ at eight symmetric/anti-symmetric locations P_1 through P_8.

mod (m,4)	Values of $\cos(m\theta)$ at locations P_1 through P_8							
	P_1 $m\theta$	P_2 $m\left(\frac{\pi}{2}-\theta\right)$	P_3 $m\left(\frac{\pi}{2}+\theta\right)$	P_4 $m(\pi-\theta)$	P_5 $m(\pi+\theta)$	P_6 $m\left(\frac{3\pi}{2}-\theta\right)$	P_7 $m\left(\frac{3\pi}{2}+\theta\right)$	P_8 $m(2\pi-\theta)$
0	$\cos(m\theta)$	$\cos(m\theta)$	$\cos(m\theta)$	$\cos(m\theta)$	$\cos(m\theta)$	$\cos(m\theta)$	$\cos(m\theta)$	$\cos(m\theta)$
1	$\cos(m\theta)$	$\sin(m\theta)$	$-\sin(m\theta)$	$-\cos(m\theta)$	$-\cos(m\theta)$	$-\sin(m\theta)$	$\sin(m\theta)$	$\cos(m\theta)$
2	$\cos(m\theta)$	$-\cos(m\theta)$	$-\cos(m\theta)$	$\cos(m\theta)$	$\cos(m\theta)$	$-\cos(m\theta)$	$-\cos(m\theta)$	$\cos(m\theta)$
3	$\cos(m\theta)$	$-\sin(m\theta)$	$\sin(m\theta)$	$-\cos(m\theta)$	$-\cos(m\theta)$	$\sin(m\theta)$	$-\sin(m\theta)$	$\cos(m\theta)$

Table 3 Values of $\sin(m\theta)$ at eight symmetric/anti-symmetric locations P_1 through P_8

mod (m,4)	Values of sin ($m\theta$)at locations P_1 through P_8							
	P_1 $m\theta$	P_2 $m\left(\frac{\pi}{2}-\theta\right)$	P_3 $m\left(\frac{\pi}{2}+\theta\right)$	P_4 $m(\pi-\theta)$	P_5 $m(\pi+\theta)$	P_6 $m\left(\frac{3\pi}{2}-\theta\right)$	P_7 $m\left(\frac{3\pi}{2}+\theta\right)$	P_8 $m(2\pi-\theta)$
0	$\sin(m\theta)$	$-\sin(m\theta)$	$\sin(m\theta)$	$-\sin(m\theta)$	$\sin(m\theta)$	$-\sin(m\theta)$	$\sin(m\theta)$	$-\sin(m\theta)$
1	$\sin(m\theta)$	$\cos(m\theta)$	$\cos(m\theta)$	$\sin(m\theta)$	$-\sin(m\theta)$	$-\cos(m\theta)$	$-\cos(m\theta)$	$-\sin(m\theta)$
2	$\sin(m\theta)$	$\sin(m\theta)$	$-\sin(m\theta)$	$-\sin(m\theta)$	$\sin(m\theta)$	$\sin(m\theta)$	$-\sin(m\theta)$	$-\sin(m\theta)$
3	$\sin(m\theta)$	$-\cos(m\theta)$	$-\cos(m\theta)$	$\sin(m\theta)$	$-\sin(m\theta)$	$\cos(m\theta)$	$\cos(m\theta)$	$-\sin(m\theta)$

Table 4 Clustering of image pixels of 8-radially symmetrical locations

mod (m,4)	F_1	F_2	F_3	F_4
0	$(f_1+f_2+f_3+f_4+f_5+f_6+f_7+f_8)$	0	$(f_1-f_2+f_3-f_4+f_5-f_6+f_7-f_8)$	0
1	$(f_1-f_4-f_5+f_8)$	$(f_2-f_3-f_6+f_7)$	$(f_2+f_3-f_6-f_7)$	$(f_1+f_4-f_5-f_8)$
2	$(f_1-f_2-f_3+f_4+f_5-f_6-f_7+f_8)$	0	$(f_1+f_2-f_3-f_4+f_5+f_6-f_7-f_8)$	0
3	$(f_1-f_4-f_5+f_8)$	$(-f_2+f_3+f_6-f_7)$	$(-f_2-f_3+f_6+f_7)$	$(f_1+f_4-f_5-f_8)$

$x_i=(2i+1)/N$, $y_k=(2k+1)/N$. After simplification, Eq. (19) becomes

The trigonometric functions $\cos(\theta_u)$ and $\sin(\theta_u)$, $u=1,2,\ldots,8$ are converted to $\cos(m\theta_{ik})$ and $\sin(m\theta_{ik})$ for various values of m according to Tables 2 and 3, respectively. The pixels are clustered in blocks represented by F_1, F_2, F_3 and F_4 expressed in pixel values f_1 through f_8 as given in Table 4. Corresponding to each pixel location in an octant there are $N(N+2)/8$ blocks. When $k=i$ there is a 4-way symmetry (along diagonals of the square image), which can be treated separately or this case can be solved using Eq. (20) itself by letting $f_2=f_4=f_6=f_8=0$.

4 Experimental results

We analyze the results by implementing three algorithms. The first algorithm implements the traditional algorithm of PHTs. The second algorithm uses the 8-way symmetry/anti-symmetry properties of the radial and angular functions [4]. The third algorithm, proposed in this paper, uses recursion for the kernel functions and clusters the image pixels and uses the symmetry/anti-symmetry properties. These algorithms are called algorithm A, B, and C, respectively. Clearly algorithm C will provide the improvement in speed performance over the existing fast method, represented by the algorithm B. These algorithms are implemented in Visual C ++ 6.0 under Microsoft Windows environment on a PC with 3.0 GHz CPU and 2 GB RAM. Since the CPU elapse time depends on the size of the image N, not on image contents, we do not mention any particular image in our experimental results. Further, the kernel functions of PCET, PCT and PST have similar forms; the results are presented for PCET only. The qualitative trend of CPU elapse time will be the same for PCTs and PSTs.

The growth of the CPU elapse time for the transform order n and repetition m is shown in Fig. 3a and b for $N=256$. Figure 3a depicts CPU elapse time for the three algorithms. To highlight differences in CPU elapse time between Algorithm B and Algorithm C, Fig. 3b is used. The results show that the proposed algorithm reduces the CPU elapse time for all orders. The quantitative values of the CPU time are given in Table 5. The proposed algorithm is faster by a factor of three to four which is evident from the

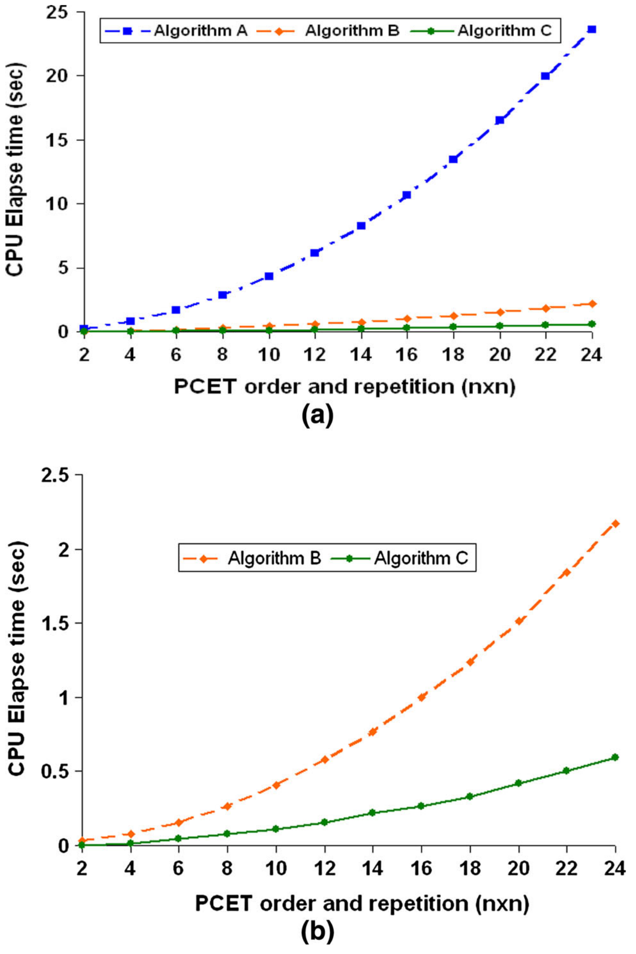

Fig. 3 CPU elapse time (s) for the computation of PCETs for image resolution 256 × 256 pixels and various orders and repetitions of transforms from $n_{\max} = m_{\max} = 2$ through $n_{\max} = m_{\max} = 24$, **a** comparison among Algorithms A, B, and C, and **b** comparison between Algorithms B, and C

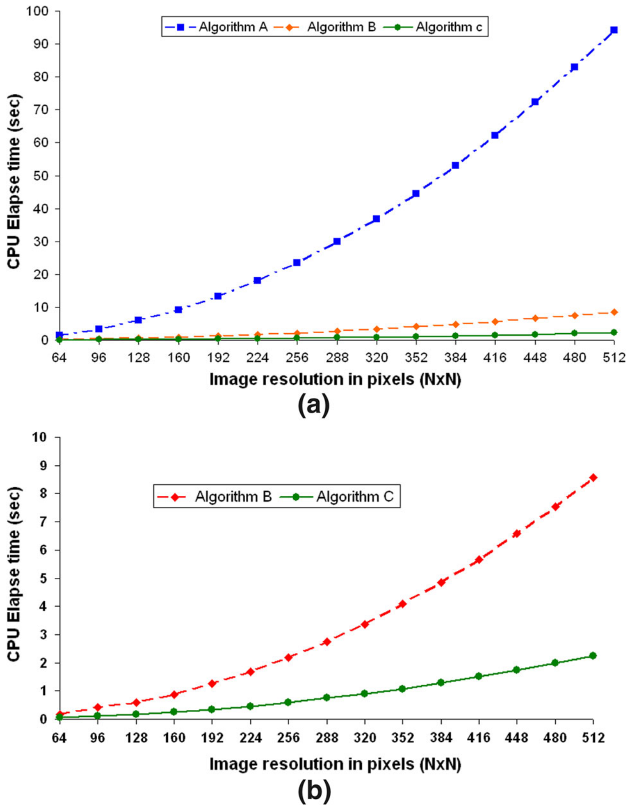

Fig. 4 CPU elapse time (s) for the computation of PCETs for fixed order and repetition $n_{\max} = m_{\max} = 24$ and various resolutions from 64 × 64 pixels through 512 × 512 pixels, **a** comparison among Algorithms A, B, and C, and **b** comparison between Algorithms B, and C

table. This is a significant improvement over the 8-way symmetry/anti-symmetry property represented by Algorithm B. To show the variation of the CPU elapse time w.r.t the size

Table 5 CPU elapse time (s) for the computation of PCET for Algorithms A, B, and C for image resolution 256 × 256 pixels and for various order and repetition of transforms from $n_{\max} = m_{\max} = 2$ through $n_{\max} = m_{\max} = 24$

PCET order n and repetition m ($n = m$)	Algorithm A (traditional approach) (s)	Algorithm B (8-way symmetry/ anti-symmetry)	Algorithm C (proposed method) (s)	Percentage saving in time by Algorithm C w.r.t Algorithm B
2	0.205	0.032	0.001	96.875
4	0.797	0.078	0.016	79.487
6	1.672	0.156	0.047	69.872
8	2.828	0.266	0.078	70.677
10	4.328	0.406	0.109	73.153
12	6.125	0.578	0.157	72.837
14	8.250	0.766	0.219	71.409
16	10.688	1.000	0.265	73.500
18	13.453	1. 235	0.329	73.360
20	16.531	1.516	0.422	72.164
22	19.953	1.843	0.500	72.870
24	23.578	2.172	0.593	72.698

of the image, we consider various values of N ranging from 64 through 512. The results are shown in Fig. 4a and b, where Fig. 4a depicts the CPU elapse time for all three algorithms and Fig. 4b highlights the difference between Algorithm B and Algorithm C. The proposed method is much faster than the existing fast method.

5 Conclusion

A fast method is developed in this paper for the calculation of PHTs, which uses recursion for the evaluation of the sinusoidal functions that comprise the kernel of the transforms. Additional enhancement in speed is achieved by clustering the image pixels that lie at eight radially symmetric locations of an octant of an image. The results of the algorithm are compared with an existing fast method, which uses 8-way symmetry/anti-symmetry property of the radial and angular parts of the kernel function. The proposed method provides a speed enhancement by a factor of three to four, a significant improvement over the existing fast method. This shows that the proposed method is suitable for applications where PHT coefficients are used as features in real-time environment involving large databases or on devices with low computation power.

Acknowledgments The authors are thankful to the anonymous reviewers for providing useful comments and suggestions for raising the standard of the paper.

References

1. Teh, C.H., Chin, R.T.: On image analysis by the methods of moments. IEEE Trans. Pattern Anal. Mach. Intell. **10**(4), 496–513 (1988)
2. Sheng, Y., Shen, L.: OFMMs for invariant pattern recognition. J. Opt. Soc. Am. A **11**((6), 1748–1757 (1994)
3. Yap, P.T., Jiang, X., Kot, A.C.: Two dimensional PHTs for invariant image representation. IEEE Trans. Pattern Anal. Mach. Intell. **32**(7), 1259–1270 (2010)
4. Yang, Z., Kamata, S.: Fast polar harmonic transforms. In: Proceedings of 11th International Conference Control, Automation, Robotics and Vision, Singapore, 7–10th December 2010, 673–676
5. Liu, M., Jiang, X., Kot, A.C., Yap, P.T.: Application of polar harmonic transforms to fingerprint classification. Emerging Topics in Computer Vision and its Applications. World Scientific Publishing, Singapore (2011)
6. Li, L., Li, S., Wang, G., Abraham, A.: An evaluation on circularly orthogonal moments for image representation. In: Proceedings of International Conference of International Science and Technology, Nanjing, Jiangsu, China, March 26–28, 394–397 (2011)
7. Miao, Q., Liu, J., Li, W., Shi, J., Wang, Y.: Three novel invariant moments based on radon and polar harmonic transform. Opt. Commun. **285**, 1044–1048 (2012)
8. Hoang, T.V., Tabbone, S.: Generic polar harmonic transforms for invariant image description. In: Proceedings of IEEE International Conference on Image Processing-ICIP 2011, Brussels, Belgium, 845–848 (2011)
9. Singh, C., Pooja, S., Upneja, R.: On image reconstruction, numerical stability, and invariance of orthogonal radial moments and radial harmonic transform. Pattern Recognit. Image Anal. **21**(4), 663–676 (2011)
10. Xin, Y., Pawlak, M., Liao, S.: Accurate Computation of Zernike moments in polar coordinates. IEEE Trans. Image Process. **16**(2), 581–587 (2007)
11. Singh, C., Walia, E.: Computation of Zernike moments in improved polar configuration. IET Image Process. **3**(4), 217–227 (2009)
12. Singh, C., Walia, E.: Algorithms for fast computation of Zernike moments and their numerical stability. Image Vis. Comput. **29**, 251–259 (2011)

Author Biographies

Chandan Singh received undergraduate degree in science in 1975 and postgraduate degree in mathematics in 1977 both from Kumaon University, Nainital, India, and Ph.D. degree in applied mathematics from Indian Institute of Technology, Kanpur, India, in 1982. He joined M/S Jyoti Ltd., Baroda, India, in1982, and later Thapar Corporate R&D Centre, Patiala, India, in 1987. In the year 1994, he joined Department of Computer Science at Punjabi University, Patiala where he is serving as Professor since 1995. In addition, he is working as Dean, Research since June 2010. Dr. Singh also served as Dean, Faculty of Engineering and Technology, from 1995 to 2000 and Dean, Faculty of Physical Sciences, during 2007–2008. He has worked in many diverse areas such as fluid dynamics, finite element analysis, optimization and numerical analysis. He has more than 35 years of teaching/ research experience. For the last 15 years he has been working in pattern recognition, optical character recognition and computer graphics. He has published more than 50 papers in various international journals and more than 45 papers in various national and international conferences.

Amandeep Kaur received B.E. degree in Electronic Engineering from Nagpur University, Nagpur, India, in 1996 and M.E. degree with honors in Computer Science and Engineering from Thapar Institute of Engineering and Technology, Patiala, India, in 2001. She was with Centre for Development of Advance Computing (C-DAC), Mohali, India, a premiere research and development institute under the Government of India from 2001 to 2005. Since 2005, she has been a lecturer in the Department of Computer Science, Punjabi University, Patiala. Her research interests include image processing and analysis. She has published eight research papers in various international journals and more than ten papers in national and international conferences.