HAL
open science

Fast Computation of Orthogonal Polar Harmonic Transforms

Thai V. Hoang, Salvatore Tabbone

To cite this version:

Thai V. Hoang, Salvatore Tabbone. Fast Computation of Orthogonal Polar Harmonic Transforms. ICPR 2012 - The 21st International Conference on Pattern Recognition, Nov 2012, Tsukuba Science City, Japan. hal-00734307

HAL Id: hal-00734307

https://inria.hal.science/hal-00734307v1

Submitted on 27 Oct 2012

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Fast Computation of Orthogonal Polar Harmonic Transforms

Thai V. Hoang[1], Salvatore Tabbone[2]
[1]*INRIA Nancy – Grand-Est, 54600 Villers-les-Nancy, France*
[2]*LORIA, University of Lorraine, 54506 Vandoeuvre-les-Nancy, France*

Abstract

This paper presents a method for the computation of polar harmonic transforms that is fast and efficient. The method is based on the inherent recurrence relations among harmonic functions that are used in the definitions of the radial and angular kernels of the transforms. The employment of these relations leads to recursive strategies for fast computation of harmonic function-based kernels. Polar harmonic transforms were recently proposed and have shown nice properties for image representation and pattern recognition. The proposed method is 10-time faster than direct computation and five-time faster than fast computation of Zernike moments.

1. Introduction

Image moments extracted from a unit disk region are usually used in invariant pattern recognition problems as rotation-invariant descriptors [13]. Let f be the image function, its moments H_{nm} over the unit disk region are computed as

$$H_{nm} = \iint\limits_{x^2+y^2\leq 1} f(x,y)V^*_{nm}(x,y)\,\mathrm{d}x\mathrm{d}y = \int_0^{2\pi}\int_0^1 f(r,\theta)V^*_{nm}(r,\theta)r\,\mathrm{d}r\mathrm{d}\theta,$$

where V_{nm} are the decomposing kernel functions and the asterisk denotes the complex conjugate. As shown in [2], the "invariance in form" preference on the computed moments H_{nm} leads to the following condition on V_{nm}:

$$V_{nm}(r,\theta) = R_n(r)A_m(\theta),$$

with $A_m(\theta) = \mathrm{e}^{im\theta}$ and $R_n(r)$ could be of any form. For example, rotational moments (RM) [14] and complex moments (CM) [1] are defined by using $R_n(r) = r^n$; angular radial transform (ART) [3] uses $R_n(r) = 2\cos(\pi nr)$. However, the obtained kernels V_{nm} of RM, CM, and ART are not orthogonal and, as a result, information redundancy exists in the moments H_{nm}, leading to difficulties in image reconstruction and low accuracy in pattern recognition. Undoubtedly, orthogonality between kernels V_{nm} comes as a natural solution to these problems. Orthogonality between the kernels means

$$\langle V_{nm}, V_{n'm'}\rangle = \delta_{nn'}\delta_{mm'} = \int_0^1 R_n(r)R^*_{n'}(r)r\,\mathrm{d}r\int_0^{2\pi} A_m(\theta)A^*_{m'}(\theta)\,\mathrm{d}\theta.$$

From the orthogonality between the angular kernels:

$$\int_0^{2\pi} A_m(\theta)A^*_{m'}(\theta)\,\mathrm{d}\theta = 2\pi\delta_{mm'},$$

the remaining condition for the radial kernels is

$$\int_0^1 R_n(r)R^*_{n'}(r)r\,\mathrm{d}r = \frac{1}{2\pi}\,\delta_{nn'}.$$

The above equation is the regulating condition for the definition of a set of radial kernels R_n in order to have orthogonality between kernels V_{nm}. There exists a number of methods that satisfies the above condition. One direction is to employ polynomials of the variable r for R_n, which turn out to be special cases of the Jacobi polynomials [10]. Popular methods are Zernike moments (ZM) [9], pseudo-Zernike moments (PZM) [14], orthogonal Fourier–Mellin moments (OFMM) [12] (a comprehensive survey could be found in [6, Section 6.3]). This class of orthogonal moments, despite its popularity, involves the computation of factorial terms, resulting in high computational complexity and numerical instability, which often limit their practical usefulness.

Another direction is to use harmonic functions for R_n as in [11] by defining

$$R_n(r) = \begin{cases} \sqrt{\frac{1}{r}}, & n = 0 \\ \sqrt{\frac{2}{r}}\sin(\pi(n+1)r), & n \text{ odd} \\ \sqrt{\frac{2}{r}}\cos(\pi nr), & n \text{ even} > 0 \end{cases} \quad (1)$$

for radial harmonic Fourier moments (RFHM) or as in [15] by defining $R_n(r) = \mathrm{e}^{i2\pi nr^2}$ for polar harmonic transforms (PCET). The radial kernels of RFHM in Eq. (1) are actually equivalent to $R_n(r) = \frac{1}{\sqrt{r}}\mathrm{e}^{i2\pi nr}$ in terms of image description, similar to the equivalence between different forms of Fourier series (i.e., trigonometric or complex exponential). This insight has led to the proposal of generic polar harmonic transform (GPCET) in [7] by generalizing the radial kernels of RFHM and PCET using a parameter s as $R_{ns}(r) = \sqrt{\frac{sr^{s-2}}{2\pi}}\,\mathrm{e}^{i2\pi nr^s}$. GPCET has been shown to have some beneficial properties for image

representation and pattern recognition, leading to comparable results with Zernike moments in these tasks. This is due to the ability of GPCET to control the distribution of the zeros of R_{ns} in the interval $0 \leq r \leq 1$. It is thus anticipated that GCPET could be a promising replacement of Jacobi polynomial-based methods in image analysis and pattern recognition tasks in the coming years.

This paper describes a method for fast computation of GPCET's radial kernels in order to further support GPCET in terms of computation complexity. The method is based on the inherent recurrence relations among harmonic functions that are used in the definitions of the radial and angular kernels of GPCET. The employment of these relations leads to recursive strategies for fast computation of harmonic function-based kernels. Since there exists no fast computation method for these newly developed transforms, the proposed method has been compared with direct computation and, in terms of kernel computation time, the best strategy is, on average, 10-time faster. In addition, when compared with the best strategy for recursive computation of Zernike moment's kernels, the proposed strategy is five-time faster.

The remainder of this paper is organized as follows. Section 2 represents the recurrence relations among harmonic functions and then proposes the recursive strategies for their fast computation. Experimental results are given in Section 3, and conclusions are drawn in Section 4.

2. Recursive computation of exponentiation

In GPCET, both the radial and angular kernels are defined based on complex exponential functions. Direct computation of exponentiation is time-consuming and often constitutes the dominant part of algorithms due to its $O(\log^2 n)$ complexity, where n refers to the number of precision digits at which the function is to be evaluated [4]. The overall complexity may become excessively high when a large number of moments is needed, or the image has high resolution, or a high-precision computation is required. Since these requirements are common in real applications, the existence of strategies for fast computation of kernels is vital for the applicability of GPCET. Fortunately, due to the following recursive definition of exponentiation:

$$\text{base case:} \quad \mathrm{e}^{i0\alpha} = 1,$$

$$\text{inductive clause:} \quad \mathrm{e}^{ik\alpha} = \mathrm{e}^{i(k-1)\alpha}\,\mathrm{e}^{i\alpha}, \quad k, \alpha \in \mathbb{Z},$$

the complex exponential functions in the definitions of GPCET radial and angular kernels can be computed recursively as

$$\mathrm{e}^{i2\pi n r^s} = \mathrm{e}^{i2\pi(n-1)r^s}\,\mathrm{e}^{i2\pi r^s},$$

$$\mathrm{e}^{im\theta} = \mathrm{e}^{i(m-1)\theta}\,\mathrm{e}^{i\theta},$$

with the base cases $\mathrm{e}^{i2\pi 0 r^s} = 1$ and $\mathrm{e}^{i0\theta} = 1$.

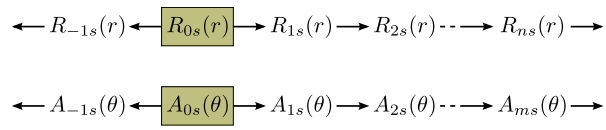


Figure 1: Recursive computation of GPCET radial kernels R_{ns} and angular kernels A_m.

Assuming $\left\{\sqrt{\frac{sr^{s-2}}{2\pi}}, \mathrm{e}^{i2\pi r^s}, \mathrm{e}^{i\theta}\right\}$ has been pre-computed and stored for polar coordinates (r, θ) of all the mapped pixel regions' centers, the following recurrence relations of R_{ns} and A_m:

$$R_{ns}(r) = \sqrt{\frac{sr^{s-2}}{2\pi}}\,\mathrm{e}^{i2\pi n r^s} = R_{(n-1)s}(r)\,\mathrm{e}^{i2\pi r^s}, \quad (2)$$

$$A_m(\theta) = \mathrm{e}^{im\theta} = A_{m-1}(\theta)\,\mathrm{e}^{i\theta}, \quad (3)$$

lead to their recursive computation with the base cases $R_{0s}(r) = \sqrt{\frac{sr^{s-2}}{2\pi}}$ and $A_0(\theta) = 1$ respectively. Obviously, computing R_{ns} from $R_{(n-1)s}$ and A_m from A_{m-1} each requires only one multiplication, which is very fast when compared to exponentiation, leading to fast computation of V_{nms}. Moreover, these forms of recurrence relations are simpler than those that were discovered for Jacobi polynomial-based radial kernels [5]. By using Eq. (2), only one recursive computational thread is sufficient to reach every GPCET radial kernels whereas many threads would be required to cover all Jacobi polynomial-based radial kernels. The computation flows of the GPCET radial kernels R_{ns} and angular kernels A_m are illustrated in Fig. 1. It is evident that the method proposed here is much faster than the one mentioned in [15] where exponentiation is required to compute complex exponential radial and angular kernels.

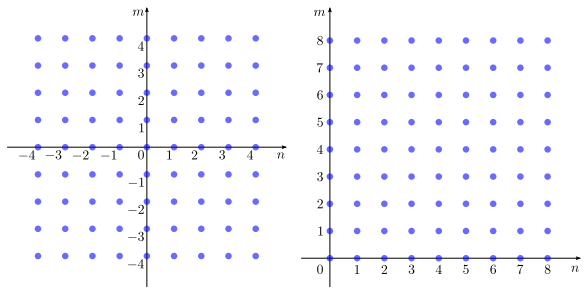
The above proposed recursive computation of GPCET radial and angular kernels could be employed for extremely fast computation of GPCET moments when the order set $\mathcal{S}$ composes a square region in $\mathbb{Z}^2$ and takes the origin as its center:

$$\mathcal{S} = \{(n, m) : n, m \in \mathbb{Z}^2, |n|, |m| \leq K\},$$

where K is a constant positive integer. By using the computational flow depicted in Fig. 2a, computing a GPCET moment thus requires only three multiplications, two for getting V_{nms} and one for multiplying V_{nms} by f, followed by a discrete sum of the obtained results over all the pixels lying entirely inside the unit disk.

In practice, to further boost the computation speed, instead of conducting the computational flow to visit all $(n, m) \in \mathcal{S}$ in four quadrants in the Cartesian space, it is sufficient to visit only $(n, m) \in \mathcal{S}$ in only one quadrant $(n, m > 0)$ as illustrated in Fig. 2b. This is possible due to the following observations:

$$R_{-ns}(r) = R^*_{ns}(r), \qquad A_{-m}(\theta) = A^*_m(\theta).$$



(a) Four quadrants (b) One quadrant

Figure 2: Computation flows of GPCET kernels.

Thus, whenever R_{ns} and A_m are available, computing the four related GPCET kernels as

$$\begin{aligned} V_{nms}(x,y) &= R_{ns}(r)A_m(\theta), \\ V_{-nms}(x,y) &= R^*_{ns}(r)A_m(\theta), \\ V_{-n-ms}(x,y) &= V^*_{nms}(x,y), \\ V_{n-ms}(x,y) &= V^*_{-nms}(x,y), \end{aligned}$$

for which eight multiplications should be needed if the computational flow in Fig. 2a is used, requires only two multiplications and three conjugations, leading to a $\frac{8}{3}$-time reduction in the number of multiplications.

3. Experimental results

The computational complexity is evaluated in terms of the elapsed time taken to compute the kernels of comparison methods from an image of size 128×128, which contains 12596 pixels lying entirely inside the unit disk. The experiments are carried out on a PC with a 2.33GHz CPU, 4GB RAM running Linux kernel $2.6.38$; MATLAB version 7.7 (R2008b) is used as the programming environment. Let K be some integer constant, all the kernels of orders (n, m) satisfying the conditions in Table 1 are computed and the averaged elapsed time over all feasible orders is taken as the kernel computation time at that value of K. The value of K is varied in the range $0 \leq K \leq 20$ in all experiments on computational complexity in order to study the trends in the dependance of kernel computation time on kernel orders. In addition, for more reliable results, all the running times indicated in this subsection are the averaged values over 100 trials.

Fig. 3 provides the computation time per kernel in milliseconds of GPCET, ZM, PZM, and OFMM. The kernels of these comparison methods are computed directly from their corresponding definition, no recursive strategy is used. It is observed from the figure that ZM, PZM, and OFMM have kernel computation time that increases almost linearly with the increase in K, meaning that a longer time is needed to compute kernels of a higher order. This is because of the evaluation of factorials of larger integers and the computation of more additive terms in the final summation. Among these methods, OFMM has the highest complexity while ZM has the lowest. This relative complexity ranking of these methods is consistent with the ranking in the number of multiplications required to compute their radial kernels.

Table 1: The constraints on the moment orders (n, m).

Moment	Order range
ZM	$\lvert m\rvert \leq n \leq K, n - \lvert m\rvert =$ even
PZM	$\lvert m\rvert \leq n \leq K$
OFMM	$0 \leq \lvert m\rvert, n \leq K$
GPCET	$\lvert m\rvert, \lvert n\rvert \leq K$

In contrast with Jacobi polynomial-based methods (ZM, PZM, and OFMM), GPCET requires almost a constant time to compute its kernels of different orders. This is due to the fact that a change in the kernel orders corresponds only to a change in the input to the complex exponential function and, as a result, this order change has almost no effect on the kernel computation time. From these observations, it can be concluded that the simple and resembling definition of harmonic function-based kernels has resulted in a fixed kernel computation time, regardless of the kernel orders. This makes a strong contrast with Jacobi polynomial-based methods where a higher order means a longer kernel computation time.

The proposed recursive strategies for fast computation of GPCET kernels are also evaluated and compared with those for fast computation of ZM kernels. The reason for comparison only to ZM is twofold: the lack of benchmarks on fast computational strategies for other methods and the popularity of recursive strategies for ZM kernels in the literature. In this comparison, ZM kernels are computed using the current state-of-the-art q-recursive strategy [5]. The comparison results are given in Fig. 4 where the legends *recursiveFour* and *recursiveOne* denote recursive computation of GPCET kernels using the computational flows in Figs. 2a and 2b respectively. It is observed that ZM kernel computation time by q-recursive increases in the range $K < 5$ and gradually decreases when $K > 5$. This is because the q-recursive strategy, which requires the pre-computation of the radial kernels R_{nn} and R_{nn-2} for each order n, is applicable only when $K \geq 4$ and is profitable when $K \geq 6$. Moreover, with the increase in n, more radial kernels R_{nm} are to be computed by recursion, leading to a decrease in the proportion of directly computed radial kernels, meaning a decrease in the averaged computation time of radial kernels.

On the other hand, using *recursiveFour* and *recursiveOne* to compute GPCET kernels leads to almost a constant computation time where *recursiveOne* is almost two-time faster than *recursiveFour*. The only exception

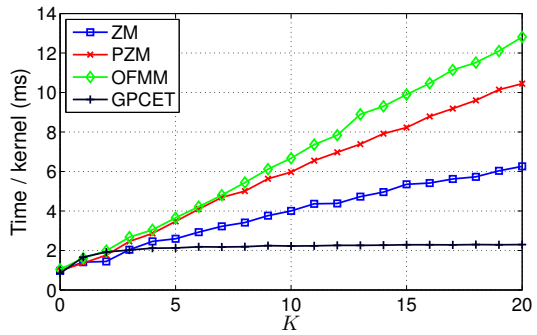


Figure 3: Direct computation.

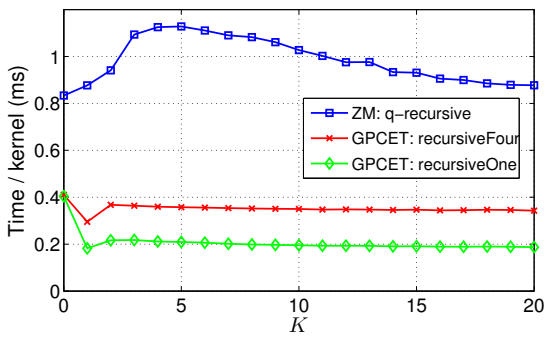


Figure 4: Recursive computation.

is at $K = 1$ where the computation time is suddenly lower. This is because MATLAB optimizes by simply copying the pre-computed values of $e^{i\theta}$ into $A_1(\theta)$ since $A_0(\theta) = 1$, instead of the more complex multiplication. Constant computation time is due to the fact that the recurrence relations in Eqs. (2) and (3) do not depend on the kernel orders and there is no need for the pre-computation of GPCET radial kernels of any order as in the ZM's q-recursive strategy. The "pure" recursive computation of radial and angular kernels is a distinct characteristic of harmonic function-based methods.

Taking *recursiveOne* as the selected strategy for recursive fast computation of GPCET kernels, recursive computation of GPCET kernels is, on average, approximately 10-time faster than direct computation of GPCET kernels and five-time faster than recursive computation of ZM kernels by q-recursive. This observation leads to a conclusion that GPCET is a promising replacement of ZM in image analysis applications where low computational complexity is an important method-selection criteria.

4. Conclusions

This paper has presented a method for the computation of polar harmonic transforms that is fast and efficient. Recursive computation strategies have been proposed by exploiting the recurrence relations among harmonic functions, leading to a method that is approximately 10-time faster than direct computation. Moreover, when compared with the current state-of-the-art strategy for fast computation of ZM kernels, the proposed method is also approximately five-time faster. Future work will investigate the combination of the proposed recursive strategies with the strategy based on geometrical symmetry [8] to multiply the computational gain obtained individually by each strategy.

References

[1] Y. Abu-Mostafa and D. Psaltis. Recognitive aspects of moment invariants. *IEEE Trans. Pattern Anal. Mach. Intell.*, 6(6):698–706, 1984.

[2] A. B. Bhatia and E. Wolf. On the circle polynomials of Zernike and related orthogonal sets. *Mathematical Proceedings of the Cambridge Philosophical Society*, 50:40–48, 1954.

[3] M. Bober. MPEG-7 visual shape descriptors. *IEEE Trans. Circuits Syst. Video Techn.*, 11(6):716–719, 2001.

[4] J. M. Borwein and P. B. Borwein. On the complexity of familiar functions and numbers. *SIAM Review*, 30(4):589–601, 1988.

[5] C.-W. Chong, P. Raveendran, and R. Mukundan. A comparative analysis of algorithms for fast computation of Zernike moments. *Pattern Rec.*, 36(3):731–742, 2003.

[6] J. Flusser, T. Suk, and B. Zitová. *Moments and Moment Invariants in Pattern Recognition*. John W.&S., 2009.

[7] T. V. Hoang and S. Tabbone. Generic polar harmonic transforms for invariant image description. In *Proceedings of ICIP*, pages 845–848, 2011.

[8] S.-K. Hwang and W.-Y. Kim. A novel approach to the fast computation of Zernike moments. *Pattern Recognition*, 39(11):2065–2076, 2006.

[9] A. Khotanzad and Y. H. Hong. Invariant image recognition by Zernike moments. *IEEE Trans. Pattern Anal. Mach. Intell.*, 12(5):489–497, 1990.

[10] Z. Ping, H. Ren, J. Zou, Y. Sheng, and W. Bo. Generic orthogonal moments: Jacobi–Fourier moments for invariant image description. *Pattern Recognition*, 40(4):1245–1254, 2007.

[11] H. Ren, Z. Ping, W. Bo, W. Wu, and Y. Sheng. Multidistortion-invariant image recognition with radial harmonic Fourier moments. *J. Opt. Soc. Am. A*, 20(4):631–637, 2003.

[12] Y. Sheng and L. Shen. Orthogonal Fourier–Mellin moments for invariant pattern recognition. *J. Opt. Soc. Am. A*, 11(6):1748–1757, 1994.

[13] M. R. Teague. Image analysis via the general theory of moments. *J. Opt. Soc. Am.*, 70(8):920–930, 1980.

[14] C.-H. Teh and R. T. Chin. On image analysis by the methods of moments. *IEEE Trans. Pattern Anal. Mach. Intell.*, 10(4):496–513, 1988.

[15] P.-T. Yap, X. Jiang, and A. C. Kot. Two-dimensional polar harmonic transforms for invariant image representation. *IEEE Trans. PAMI*, 32(6):1259–1270, 2010.