

Fast Hypercomplex Polar Fourier Analysis for Image Processing

Zhuo Yang and Sei-ichiro Kamata

Graduate School of Information, Production and Systems
Waseda University, Japan
joel@ruri.waseda.jp, kam@waseda.jp

Abstract. Hypercomplex polar Fourier analysis treats a signal as a vector field and generalizes the conventional polar Fourier analysis. It can handle signals represented by hypercomplex numbers such as color images. It is reversible that can reconstruct image. Its coefficient has rotation invariance property that can be used for feature extraction. With these properties, it can be used for image processing applications like image representation and image understanding. However in order to increase the computation speed, fast algorithm is needed especially for image processing applications like realtime systems and limited resource platforms. This paper presents fast hypercomplex polar Fourier analysis that based on symmetric properties and mathematical properties of trigonometric functions. Proposed fast hypercomplex polar Fourier analysis computes symmetric eight points simultaneously that significantly reduce the computation time.

Keywords: fast hypercomplex polar Fourier analysis, hypercomplex polar Fourier analysis, Fourier analysis.

1 Introduction

Fourier transforms have been widely used in image processing, signal processing and many engineering fields [1]. By representing image as hypercomplex numbers, especially the quaternions discovered by Hamilton [2], HyperComplex Fourier transform is proposed as generalization of quaternion Fourier transform for color image processing [3]. The relationship between right-side quaternion Fourier transform and left-side quaternion Fourier transform is established [4]. Based on hypercomplex Fourier transform, effective algorithms for motion estimation in color image sequences are studied [5]. Quaternionic Gabor filters are designed to combine the color channels and the orientations in the image plane [6].

Inspired from these, hypercomplex polar Fourier analysis is studied. By introducing a hypercomplex number, hypercomplex polar Fourier analysis [8] treats a signal as a vector field and generalizes the polar Fourier analysis [7]. Hypercomplex polar Fourier analysis can handle color image. With orthogonality, it can decompose and reconstruct color image. The coefficients hold rotation invariant property. With these properties, it can be widely used as an image

processing tool. Unfortunately, the computations of hypercomplex polar Fourier analysis involve many Bessel function and trigonometric computations that no fast method has been reported. Therefore, reduction of the computation time is very significant.

This paper focuses on fast hypercomplex polar Fourier analysis. Fast and compact method to compute the coefficients of hypercomplex polar Fourier analysis is proposed by using mathematical properties of trigonometric functions and points relationships. The two dimensional basis function of hypercomplex polar Fourier analysis has symmetry properties with respect to the x axis, y axis, $y = x$ line, $y = -x$ line and origin that can be used for fast computation. The computational complexity can be reduced by calculating half of the first quadrant. For image processing applications, computation time is important factor. Using the proposed method, only one eighth is needed compared with the direct calculation.

The organization of this paper is as follows. The basic theory of hypercomplex polar Fourier analysis including mathematics definitions are provided in Section 2. Section 3 presents the proposed method in detail. Experiments are designed to demonstrate effectiveness of the proposed method in Section 4. Finally, Section 5 concludes this study.

2 Background

2.1 Hypercomplex Number

As a type of hypercomplex number and generalization of complex number, the quaternion, its properties and applications have been studied [9]. In signal and image processing, quaternion number based methods are actively researched. Such as, quaternionic Gabor filters are designed to combine the color channels and the orientations in the image plane [6]. Quaternionic phase correlation based motion estimation approach is studied [5].

Complex number has two components, the real part and imaginary part. Quaternion has one real part and three imaginary parts. Given $a, b, c, d \in \mathbb{R}$, a quaternion $q \in \mathbb{H}$ (\mathbb{H} denotes Hamilton) is defined as

$$q = \mathcal{S}(q) + \mathcal{V}(q), \mathcal{S}(q) = a, \mathcal{V}(q) = bi + cj + dk \quad (1)$$

where $\mathcal{S}(q)$ is scalar part and $\mathcal{V}(q)$ is vector part. i, j, k are imaginary operators obeying the following rules

$$\begin{aligned} i^2 = j^2 = k^2 &= -1, ij = -ji = k, \\ jk &= -kj = i, ki = -ik = j, \end{aligned} \quad (2)$$

The norm of quaternion q is

$$\|q\| = \sqrt{a^2 + b^2 + c^2 + d^2}. \quad (3)$$

Quaternion q is named as unit quaternion if it is in set

$$\mathbb{U} = \{q | q \in \mathbb{H}, \|q\| = 1\}. \quad (4)$$

If quaternion q in following set,

$$\mathbb{P} = \{q|q \in \mathbb{H}, \mathcal{S}(q) = 0\}, \tag{5}$$

it is called pure quaternion. The quaternions belonging to set

$$\mathbb{S} = \{q|q \in \mathbb{U}, q \in \mathbb{P}\}, \tag{6}$$

are called unit pure quaternion. Euler formula holds for hypercomplex numbers,

$$e^{\mu\phi} = \cos(\phi) + \mu \sin(\phi) \tag{7}$$

2.2 HyperComplex Polar Fourier Analysis

Given a 2D function $f(x, y)$, it can be transformed from cartesian coordinate to polar coordinate $f(r, \varphi)$, where r and φ denote radius and azimuth respectively. The following equations transform from cartesian coordinate to polar coordinate,

$$r = \sqrt{x^2 + y^2}, \tag{8}$$

and

$$\varphi = \arctan \frac{y}{x}. \tag{9}$$

Hypercomplex Polar Fourier analysis involves points within the largest inner circle of the image. After normalization, it is defined on the unit circle that $r \leq 1$ and can be expanded with respect to the basis function. Hypercomplex polar Fourier analysis is defined as

$$f(r, \varphi) = \sum_{n=1}^{\infty} \sum_{m=-\infty}^{\infty} HP_{nm} R_{nm}(r) e^{\mu m \varphi}, \tag{10}$$

where μ is unit pure quaternion and is defined as $\mu = \frac{1}{\sqrt{3}}i + \frac{1}{\sqrt{3}}j + \frac{1}{\sqrt{3}}k$, and the coefficient is

$$\begin{aligned} HP_{nm} &= \frac{1}{\sqrt{2\pi}} \int_0^1 \int_0^{2\pi} R_{nm}(r) f(r, \varphi) e^{-\mu m \varphi} r dr d\varphi \\ &= \frac{1}{\sqrt{2\pi}} \int_0^1 \int_0^{2\pi} R_{nm}(r) f(r, \varphi) (\cos m\varphi - \mu \sin m\varphi) r dr d\varphi \end{aligned} \tag{11}$$

where

$$R_{nm}(r) = \frac{1}{\sqrt{N_n^{(m)}}} J_m(x_{mn}r), \tag{12}$$

in which J_m is the m-th order first class Bessel series [10], and $N_n^{(m)}$ can be deduced by imposing boundary conditions according to the Sturm-Liouville(S-L) theory [11]. With zero-value boundary condition,

$$N_n^{(m)} = \frac{1}{2} J_{m+1}^2(x_{mn}), \tag{13}$$

in which x_{mn} is the nth positive root for $J_m(x)$.

The coefficient HP_{nm} is rotation invariant. Hypercomplex polar Fourier analysis is reversible. Fig. 1 shown that image can be reconstructed. With n increases bigger, more detail part of the image can be obtained.

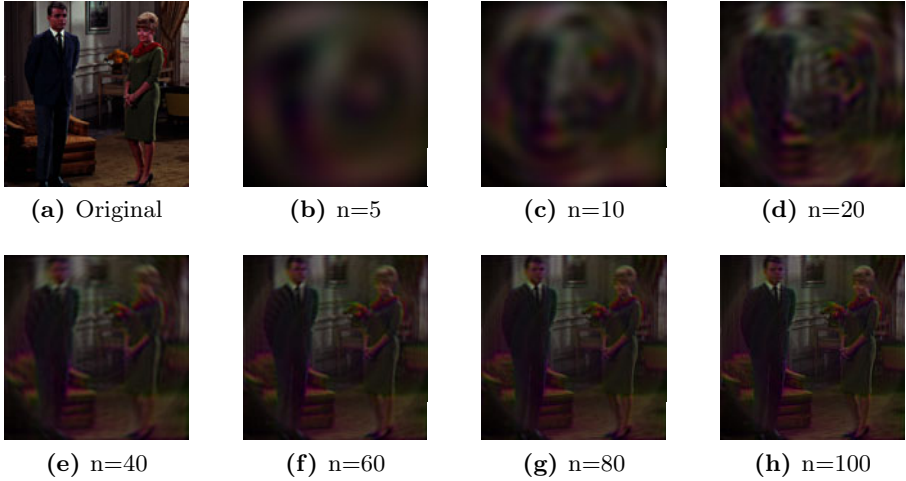


Fig. 1. Hypercomplex polar Fourier analysis

3 Fast Hypercomplex Polar Fourier Analysis

This section presents fast hypercomplex polar Fourier analysis. From Eq. 11, for same radius r , the different integrand part for each point is $f(r, \varphi)(\cos m\varphi - \mu \sin m\varphi)$. As shown in Fig.2 , point (x, y) is a point in first quadrant below $y = x$, has seven other symmetric points with respect to x axis, y axis, $y = x$, $y = -x$ and origin.

Mappings between polar and cartesian coordinates are show in Table 1.

Within period 2π , $\sin(\varphi)$ and $\cos(\varphi)$ functions are periodic functions. Periods for $\sin(m\varphi)$ and $\cos(m\varphi)$ are $2\pi/m$. Derived from the periodic and symmetric properties of trigonometric functions that used in FFT [12] , mathematical relationships for trigonometric functions exist with respect to different m . If l is divided by 4 with remainder 3 that means $\text{mod}(l, 4) = 3$, following relationship for sine function can be deduced

$$\sin(l(\frac{\pi}{2} - \theta)) = -\cos(l\theta), \tag{14}$$

$$\sin(l(\frac{\pi}{2} + \theta)) = -\cos(l\theta), \tag{15}$$

$$\sin(l(\pi - \theta)) = \sin(l\theta), \tag{16}$$

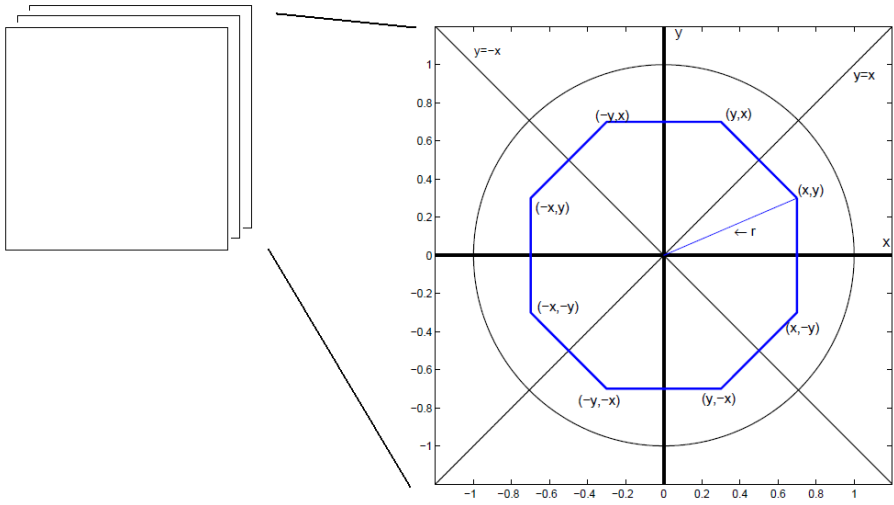


Fig. 2. Two dimensional symmetric points in multiple layers

$$\sin(l(\pi + \theta)) = -\sin(l\theta), \tag{17}$$

$$\sin(l(\frac{3\pi}{2} - \theta)) = \cos(l\theta), \tag{18}$$

$$\sin(l(\frac{3\pi}{2} + \theta)) = \cos(l\theta), \tag{19}$$

$$\sin(l(2\pi - \theta)) = -\sin(l\theta). \tag{20}$$

Similar relationships also exist for cosine function and other l values. For the eight symmetric points on the same radius r , if coefficients can be calculated simultaneously, then the computation time for trigonometric function and Bessel function can be reduced.

Based on foregoing discussion, fast hypercomplex polar Fourier analysis is given by

$$FastHP_{nm} = \frac{1}{\sqrt{2\pi}} \iint_D R_{nm}(\sqrt{x^2 + y^2}) (G_m(x, y) - \mu H_m(x, y)) dx dy, \tag{21}$$

where

$$D = \{(x, y) | 0 \leq x \leq 1, 0 \leq y \leq x, 0 \leq x^2 + y^2 \leq 1\}, \tag{22}$$

and $G_m(x, y)$ and $H_m(x, y)$ are given in followings.

With same result, proposed method can share computation between symmetric points that significantly reduce the time in order to obtain the final result. The proposed method is unrelated to image content. Experiments are designed and results are given in following section.

Table 1. (r, θ) and its symmetric points

Polar Coordinate	Cartesian Coordinate
(r, θ)	(x, y)
$(r, \frac{\pi}{2} - \theta)$	(y, x)
$(r, \frac{\pi}{2} + \theta)$	$(-y, x)$
$(r, \pi - \theta)$	$(-x, y)$
$(r, \pi + \theta)$	$(-x, -y)$
$(r, \frac{3\pi}{2} - \theta)$	$(-y, -x)$
$(r, \frac{3\pi}{2} + \theta)$	$(y, -x)$
$(r, 2\pi - \theta)$	$(x, -y)$

$$G_m(x, y) = \begin{cases} \left(\begin{array}{l} (f(x, y) + f(y, x) + f(-y, x) + f(-x, y) \\ + f(-x, -y) + f(-y, -x) + f(y, -x) + f(x, -y))\cos(m\varphi) \\ (f(x, y) - f(-x, y) - f(-x, -y) + f(x, -y))\cos(m\varphi) \\ + (f(y, x) - f(-y, x) - f(-y, -x) + f(y, -x))\sin(m\varphi) \end{array} \right) & \text{if } \text{mod}(m, 4) = 0 \\ \left(\begin{array}{l} (f(x, y) - f(y, x) - f(-y, x) + f(-x, y) \\ + f(-x, -y) - f(-y, -x) - f(y, -x) + f(x, -y))\cos(m\varphi) \\ (f(x, y) - f(-x, y) - f(-x, -y) + f(x, -y))\cos(m\varphi) \\ - (f(y, x) - f(-y, x) - f(-y, -x) + f(y, -x))\sin(m\varphi) \end{array} \right) & \text{if } \text{mod}(m, 4) = 1 \\ \left(\begin{array}{l} (f(x, y) - f(y, x) + f(-y, x) - f(-x, y) \\ + f(-x, -y) - f(-y, -x) + f(y, -x) - f(x, -y))\sin(m\varphi) \\ (f(x, y) + f(-x, y) - f(-x, -y) - f(x, -y))\sin(m\varphi) \\ + (f(y, x) + f(-y, x) - f(-y, -x) - f(y, -x))\cos(m\varphi) \end{array} \right) & \text{if } \text{mod}(m, 4) = 2 \\ \left(\begin{array}{l} (f(x, y) + f(y, x) - f(-y, x) - f(-x, y) \\ + f(-x, -y) + f(-y, -x) - f(y, -x) - f(x, -y))\sin(m\varphi) \\ (f(x, y) + f(-x, y) - f(-x, -y) - f(x, -y))\sin(m\varphi) \\ - (f(y, x) + f(-y, x) - f(-y, -x) - f(y, -x))\cos(m\varphi) \end{array} \right) & \text{if } \text{mod}(m, 4) = 3 \end{cases} \quad (23)$$

$$H_m(x, y) = \begin{cases} \left(\begin{array}{l} (f(x, y) - f(y, x) + f(-y, x) - f(-x, y) \\ + f(-x, -y) - f(-y, -x) + f(y, -x) - f(x, -y))\sin(m\varphi) \\ (f(x, y) + f(-x, y) - f(-x, -y) - f(x, -y))\sin(m\varphi) \\ + (f(y, x) + f(-y, x) - f(-y, -x) - f(y, -x))\cos(m\varphi) \end{array} \right) & \text{if } \text{mod}(m, 4) = 0 \\ \left(\begin{array}{l} (f(x, y) + f(y, x) - f(-y, x) - f(-x, y) \\ + f(-x, -y) + f(-y, -x) - f(y, -x) - f(x, -y))\sin(m\varphi) \\ (f(x, y) + f(-x, y) - f(-x, -y) - f(x, -y))\sin(m\varphi) \\ - (f(y, x) + f(-y, x) - f(-y, -x) - f(y, -x))\cos(m\varphi) \end{array} \right) & \text{if } \text{mod}(m, 4) = 1 \\ \left(\begin{array}{l} (f(x, y) - f(y, x) + f(-y, x) - f(-x, y) \\ + f(-x, -y) - f(-y, -x) + f(y, -x) - f(x, -y))\sin(m\varphi) \\ (f(x, y) + f(-x, y) - f(-x, -y) - f(x, -y))\sin(m\varphi) \\ + (f(y, x) + f(-y, x) - f(-y, -x) - f(y, -x))\cos(m\varphi) \end{array} \right) & \text{if } \text{mod}(m, 4) = 2 \\ \left(\begin{array}{l} (f(x, y) + f(y, x) - f(-y, x) - f(-x, y) \\ + f(-x, -y) + f(-y, -x) - f(y, -x) - f(x, -y))\sin(m\varphi) \\ (f(x, y) + f(-x, y) - f(-x, -y) - f(x, -y))\sin(m\varphi) \\ - (f(y, x) + f(-y, x) - f(-y, -x) - f(y, -x))\cos(m\varphi) \end{array} \right) & \text{if } \text{mod}(m, 4) = 3 \end{cases} \quad (24)$$

4 Experimental Results

The performance of the proposed fast hypercomplex polar Fourier analysis in computation reduction is validated through comparative experiments using different images. Images with different content are tested for test to illustrate the efficiency and feasibility of the proposed method over direct computation. PC

environment (Celeron 1.86GHz, 2G Memory) is used to perform the experiments. Algorithms are implemented by C++. GNU Scientific Library [13] is used for Bessel function.

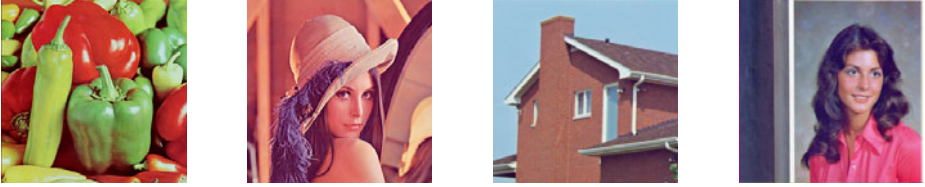


Fig. 3. Standard Images

Test data set consists of standard images as shown in Fig. 3 . With different number of coefficients computed, the performances in terms of CPU elapsed time are given in Table 2 . With same computation result, but two methods have different running time. From the result, the proposed method is effective and is unrelated to number of coefficients and image content. It takes one eighth time to compute hypercomplex polar Fourier analysis that significantly boost the speed. Applications that need hypercomplex polar Fourier analysis for image representation and image understanding can largely reduce the computation time.

Table 2. CPU Elapsed Time to Test Standard Images

Coefficients	Direct Method	Proposed Method	Ratio
5	0.965	0.130	0.135
10	1.924	0.262	0.136
20	3.888	0.532	0.137
30	5.884	0.804	0.137
40	7.846	1.066	0.136

5 Conclusions

In this paper, fast hypercomplex polar Fourier analysis is proposed. Hypercomplex polar Fourier analysis can be used in applications like image representation and image understanding as discussed in previous work. By using the symmetric properties and mathematical properties of trigonometric functions , the proposed methods only calculate one eighth of trigonometric functions and Bessel functions. That means, proposed method can largely decrease the calculation time by sharing computation between symmetric points. Experimental results are given on different images to illustrate the effectiveness. Image processing applications that need fast hypercomplex polar Fourier analysis will benefit from this work.

Acknowledgments. This work was supported in part by Grant-in-Aid (No.21500181) for Scientific Research by the Ministry of Education, Science and Culture of Japan.

References

1. Bracewell, R.N.: The Fourier transform and its applications. McGraw-Hill (2000)
2. Hamilton, W.R.: Elements of Quaternions. Longmans Green, London (1866)
3. Ell, T.A., Sangwine, S.J.: Hypercomplex Fourier Transforms of Color Images. *IEEE Trans. IP* 16(1), 22–35 (2007)
4. Yeh, M.-H.: Relationships Among Various 2-D Quaternion Fourier Transforms. *IEEE Trans. SPL* 15, 669–672 (2008)
5. Alexiadis, D.S., Sergiadis, G.D.: Motion estimation, segmentation and separation, using hypercomplex phase correlation, clustering techniques and graph-based optimization. *Computer Vision and Image Understanding* 113(2), 212–234 (2009)
6. Subakan, O.N., Vemuri, B.C.: A Quaternion Framework for Color Image Smoothing and Segmentation. *International Journal of Computer Vision* 91(3), 233–250 (2011)
7. Wang, Q., Ronneberger, O., Burkhardt, H.: Rotational Invariance Based on Fourier Analysis in Polar and Spherical Coordinates. *IEEE Trans. PAMI* 31(9), 1715–1722 (2009)
8. Yang, Z., Kamata, S.: Hypercomplex Polar Fourier Analysis for Image Representation. *IEICE Trans. on Info. and Sys.* E94-D(8), 1663–1670 (2011)
9. Ward, J.P.: Quaternions and Cayley Numbers, Algebra and Applications. Springer, Heidelberg (1997)
10. Andrews, L.: Special Functions of Mathematics for Engineers, 2nd edn. SPIE Press (1997)
11. Kosmala, W.: Advanced Calculus: a friendly approach. Prentice-Hall (1999)
12. Burrus, C.S., Parks, T.W.: DFT/FFT and Convolution Algorithms and Implementation. John Wiley & Sons (1985)
13. The Gnu Scientific Library, <http://www.gnu.org/software/gsl/>