IEEE Access
Multidisciplinary : Rapid Review : Open Access Journal

Received April 27, 2020, accepted May 13, 2020, date of publication May 18, 2020, date of current version June 2, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.2995392

GPU Accelerated Polar Harmonic Transforms for Feature Extraction in ITS Applications

ZHUO YANG[1], MINGKAI TANG[1], ZHUOZHANG LI[1], ZILIANG REN[2,3], AND QIESHI ZHANG[2,3], (Member, IEEE)

[1]School of Computers, Guangdong University of Technology, Guangzhou 510006, China
[2]CAS Key Laboratory of Human-Machine Intelligence-Synergy Systems, Shenzhen Institutes of Advanced Technology, Chinese Academy of Sciences, Shenzhen 518055, China
[3]Department of Mechanical and Automation Engineering, The Chinese University of Hong Kong, Hong Kong

Corresponding author: Qieshi Zhang (qs.zhang@siat.ac.cn)

This work was supported in part by the Natural Science Foundation of China under Grant 61907009, Grant U1913202, and Grant U1813205, in part by the Science and Technology Planning Project of Guangdong Province under Grant 2017B010110007, Grant 2017B010110015, and Grant 2019B010150002, in part by the Natural Science Foundation of Guangdong Province under Grant 2018A030313802, in part by the Guangdong-Hong Kong-Macao Joint Laboratory of Human–Machine Intelligence-Synergy Systems under Grant 2019B121205007, and in part by the Shenzhen Technology Project under Grant JCYJ20180507182610734.

ABSTRACT Polar Harmonic Transform (PHT) is termed to represent a set of transforms those kernels are basic waves and harmonic in nature, which can improve the effect in Intelligent Transportation System (ITS) applications. PHTs consist of Polar Complex Exponential Transform (PCET), Polar Cosine Transform (PCT) and Polar Sine Transform (PST). PHTs can extract orthogonal and rotation invariant features and demonstrated superior performance in various image processing and computer vision applications. For real time systems and large multimedia databases, execution efficiency is always a significant challenge. With widespread use of Graphics Processing Unit (GPU), this study presents GPU based PHTs. Proposed methods are based on mathematical properties of PHTs and optimization techniques of GPU. Optimal parameter selections for GPU execution are also discussed. In our experiments, proposed methods are over 1800 times faster.

INDEX TERMS GPU, Polar harmonic transform, feature extraction, intelligent transportation system.

I. INTRODUCTION

With rapid development of artificial intelligence, intelligent transportation system (ITS) especially autonomous driving attracts multidisciplinary researchers and becomes one of most promising directions. Autonomous driving technologies are mainly divided into three parts: perceptual positioning, planning decision making and executive control. As for perceptual positioning, there are challenging tasks including driver environment understanding [1], [2], road sign detection [3], [4], pedestrian detection [5], [6], behaviour analysis and prediction [7], depth estimation [8], [9], Vehicle-to-everything (V2X) [10]–[12], and intelligent human-computer interaction technology [13], [14]. Among these ITS applications and tasks, feature extraction plays a significant role.

Polar Harmonic Transforms (PHTs) consist of Polar Complex Exponential Transform (PCET), Polar Cosine Transform (PCT) and Polar Sine Transform (PST) [15]. PHTs can extract orthogonal and rotation invariant features and demonstrated superior performance in ITS tasks. Farnoosh and Ali [16] use PCT coefficients to correct uncertain labels for getting more accurate body reconstruction. Al-asady and Al-amery [17] obtain accurate features for human action detection. Lin *et al.* [18] and Liu *et al.* [19] propose region duplication detection scheme for feature point mapping.

PHTs also show competitive result in applications like image watermarking [20]–[24], fingerprint indexing [25], image copy-move forgery detection [26]–[29], color image analysis [30], [31], breast cancer detection [32], hand vein recognition [33], binary image recognition [34], MRI data analysis [35], [36], video hashing [37], [38], airborne platform localization [39], image retrieval [40]. For compute-intensive tasks Graphics Processing Unit (GPU) based parallel computing shows obvious advantages in many fields like Wavelet transform [41], Fourier transform [42]. Computing speed is very important for ITS applications.

This paper focuses on GPU based Polar Harmonic Transforms (GPHTs) that consist of GPU based Polar Complex Exponential Transform (GPCET), Polar Cosine Transform (GPCT) and Polar Sine Transform (GPST). For utilizing

The associate editor coordinating the review of this manuscript and approving it for publication was Edith C.-H. Ngai.

GPU parallel computational capability, we implement proposed methods with Compute Unified Device Architecture (CUDA) [43], [44]. Mathematical properties of PHTs are also considered in CUDA. Different execution configurations of GPU lead to different running time. Optimal parameter selection are evaluated as well.

The organization of this paper is as follows. The mathematics definitions of PCET, PCT and PST are given in Section 2. The proposed methods are presented in Section 3 after introducing GPU memory structure and parallel execution model. In Section 4, the performance of proposed method is evaluated under different images. The experimental results illustrate that proposed method is really effective. Finally, concludes this study.

II. POLAR HARMONIC TRANSFORMS

This section introduces PHTs, and for further details refer to [15].

A. POLAR COMPLEX EXPONENTIAL TRANSFORM (PCET)

Given a 2D image function $f(x, y)$, it can be transformed from Cartesian coordinate to polar coordinate $f(r, \theta)$ as following formulae transform, where r and θ denote radius and azimuth respectively.

$$r = \sqrt{x^2 + y^2}, \tag{1}$$

and

$$\theta = \arctan\left(\frac{y}{x}\right). \tag{2}$$

PCET is defined on the unit circle that $r \leq 1$, and can be expanded with respect to the basis functions $H_{nl}(r, \theta)$ as

$$f(r, \theta) = \sum_{n=-\infty}^{\infty} \sum_{l=-\infty}^{\infty} M_{nl} H_{nl}(r, \theta), \tag{3}$$

where the coefficient is

$$M_{nl} = \frac{1}{\pi} \int_0^{2\pi} \int_0^1 f(r, \theta) H_{nl}^*(r, \theta) r dr d\theta. \tag{4}$$

The basis function is given by

$$H_{nl}(r, \theta) = R_n(r) e^{il\theta}, \tag{5}$$

where

$$R_n(r) = e^{i2\pi nr^2}. \tag{6}$$

Rewrite Eq. (4) with Eqs. (5) and (6):

$$M_{nl} = \frac{1}{\pi} \int_0^{2\pi} \int_0^1 f(r, \theta)(\cos(2\pi nr^2 + l\theta) - i\sin(2\pi nr^2 + l\theta)) r dr d\theta, \tag{7}$$

$|M_{nl}|$ is rotation invariant and can be used for feature extraction.

B. POLAR COSINE TRANSFORM AND POLAR SINE TRANSFORM (PCT & PST)

PCT is given by

$$f(r, \theta) = \sum_{n=0}^{\infty} \sum_{l=-\infty}^{\infty} M_{nl}^C H_{nl}^C(r, \theta), \tag{8}$$

where the coefficient is

$$M_{nl}^C = \Omega_n \int_0^{2\pi} \int_0^1 f(r, \theta) H_{nl}^{C*}(r, \theta) r dr d\theta. \tag{9}$$

The basis function of PCT is

$$H_{nl}^C(r, \theta) = R_n^C(r)\, e^{il\theta}, \tag{10}$$

where

$$R_n^C(r) = \cos(\pi nr^2), \tag{11}$$

and

$$\Omega_n = \begin{cases} \frac{1}{\pi} & \text{if } n = 0 \\ \frac{2}{\pi} & \text{if } n \neq 0. \end{cases} \tag{12}$$

Rewrite Eq. (9) with Eqs. (10), (11) and (12):

$$M_{nl}^C = \Omega_n \int_0^{2\pi} \int_0^1 f(r, \theta) \cos(\pi nr^2) \times (\cos(l\theta) - i\sin(l\theta)) r dr d\theta. \tag{13}$$

Similarly, PST is given by

$$f(r, \theta) = \sum_{n=1}^{\infty} \sum_{l=-\infty}^{\infty} M_{nl}^S H_{nl}^S(r, \theta), \tag{14}$$

where the coefficient is

$$M_{nl}^S = \Omega_n \int_0^{2\pi} \int_0^1 f(r, \theta) H_{nl}^{S*}(r, \theta) r dr d\theta. \tag{15}$$

The basis function of PST is

$$H_{nl}^S(r, \theta) = R_n^S(r)\, e^{il\theta}, \tag{16}$$

where

$$R_n^S(r) = \sin(\pi nr^2), \tag{17}$$

Rewrite Eq. (15) with Eqs. (16) and (17):

$$M_{nl}^S = \Omega_n \int_0^{2\pi} \int_0^1 f(r, \theta) \sin(\pi nr^2) \times (\cos(l\theta) - i\sin(l\theta)) r dr d\theta. \tag{18}$$

PCT and PST are defined on unit circle as well. $|M_{nl}^C|$ and $|M_{nl}^S|$ are rotation invariant.

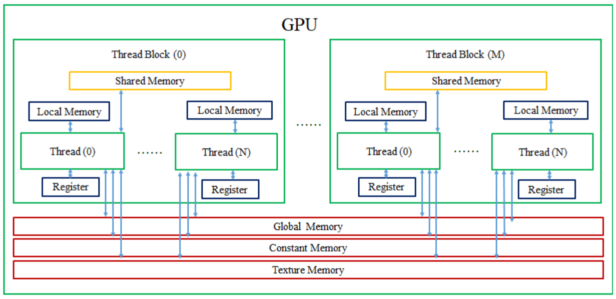


FIGURE 1. Logical view of GPU.

III. PROPOSED METHOD

A. GPU ARCHITECTURE

GPU contains Streaming Multiprocessors (SM). Parallel threads on SM are grouped into a thread block. Thread blocks are grouped into a grid that corresponds to a CUDA kernel function call in a GPU program [43]. Each block in a grid has its own block id. The number of threads in a thread block and the number of thread blocks in a thread grid can be specified and can also impact the computation efficiency [44].

GPU has several memory models such as register, local memory, shared memory, global memory, constant memory and texture memory as shown in FIGURE 1.

B. DIRECT COMPUTATION METHOD ON GPU

Each GPU thread handles a pixel data. Parallel threads significantly boost PHTs. Following variables can be set on GPU. *gridDim.x* is the number of thread blocks in a thread grid, *blockId.x* is the index of thread block in a thread grid, *blockDim.x* is the number of threads in a thread block, *threadId.x* is the index of thread in a thread block. We define *id* as the index of a thread in a thread grid. *id* can be calculated by following formula,

$$id = blockDim.x \times blockId.x + threadId.x \tag{19}$$

For an image with $N \times N$ resolution, each pixel is represented as (x, y). The pixel in x-th column and y-th row can be mapped to GPU thread *id* by following equation:

$$y \times N + x = id, \tag{20}$$

where $0 \leq y < N$, $0 \leq x < N$. x and y can be calculated from id:

$$y = \left\lfloor \frac{id}{N} \right\rfloor, \tag{21}$$

$$x = mod(id, N), \tag{22}$$

where $mod(\cdot)$ is modulo operator. According to Eqs. (21) and (22), parallel GPU threads with different *id* can access different (x, y) pixel data to accomplish PHTs on GPU directly.

C. FAST COMPUTATION METHOD ON GPU

From Eq. (13), we can find for the pixels with same r and $\cos(\pi n r^2)$, the different integrated part is

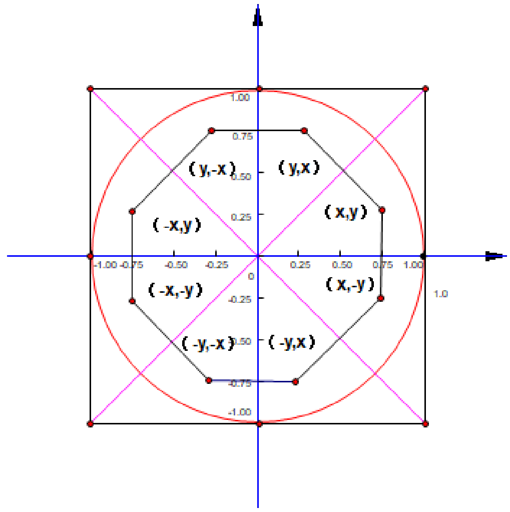


FIGURE 2. Symmetric pixels as a group.

$f(r,\theta)(\cos(l\theta) - i\sin(l\theta))$. Axis symmetric and origin symmetric pixels like (x,y), $(-x,y)$, $(x,-y)$, $(-x,-y)$, (y,x), $(-y,x)$, $(y,-x)$, $(-y,-x)$ can be grouped and share computation as shown in FIGURE 2. Their Cartesian and polar coordinates are shown in Table 1.

As known $\sin(\theta)$ and $\cos(\theta)$ functions are periodic functions with period 2π. Periods for $\sin(l\theta)$ and $\cos(l\theta)$ are $2\pi/l$. Derived from the periodic and symmetric properties of trigonometric functions that used in Fast Fourier Transform (FFT) [45], mathematical relationships for trigonometric functions exist with respect to different l. If l is divided by 4 with remainder 1 that means $mod(l, 4) = 1$, following relationship for sine function can be deduced:

$$\sin(l(\frac{\pi}{2} - \theta)) = \cos(l\theta), \tag{23}$$

$$\sin(l(\frac{\pi}{2} + \theta)) = \cos(l\theta), \tag{24}$$

$$\sin(l(\pi - \theta)) = \sin(l\theta), \tag{25}$$

$$\sin(l(\pi + \theta)) = -\sin(l\theta), \tag{26}$$

$$\sin(l(\frac{3\pi}{2} - \theta)) = -\cos(l\theta), \tag{27}$$

$$\sin(l(\frac{3\pi}{2} + \theta)) = -\cos(l\theta), \tag{28}$$

$$\sin(l(2\pi - \theta)) = -\sin(l\theta). \tag{29}$$

Similar relationships also exist for cosine function and other l values. For the eight symmetric points on the same radius r, coefficients can be calculated simultaneously.

Based on previous discussion, we rewrite Eq. (13) and have GPCT

$$GPUM_{nl}^{C} = \Omega_n \iint_D w(x,y)\cos(\pi n(x^2+y^2)) \times (G_l(x,y) - iH_l(x,y))\,dx\,dy, \tag{30}$$

where

$$D = \{(x,y) | 0 \leq x \leq 1, 0 \leq y \leq x, 0 \leq x^2 + y^2 \leq 1\}, \tag{31}$$

TABLE 1. Coordinates for symmetric pixels.

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

where $G_l(x, y)$ and $H_l(x, y)$ is shown in Eqs. (32) and (33), as shown at the bottom of this page, and $w(x, y)$ is given by

$$w(x, y) = \begin{cases} 1 & if\ (x, y) \notin P \\ \frac{1}{2} & if\ (x, y) \in P, \end{cases} \tag{34}$$

where

$$P = \{(x, y)|y = x,\ y = -x,\ x = 0,\ y = 0\}. \tag{35}$$

Similarly, GPST is given by

$$GPUM^S_{nl} = \Omega_n \iint_D w(x, y) \sin(\pi n(x^2 + y^2)) \times (G_l(x, y) - iH_l(x, y))\, dx\, dy. \tag{36}$$

As for PCET, we can simplify rewrite Eq. (7) based mathematical property of trigonometric functions as,

$$\cos(2\pi nr^2 + l\theta) = \cos(2\pi nr^2)\cos(l\theta) - \sin(2\pi nr^2)\sin(l\theta), \tag{37}$$

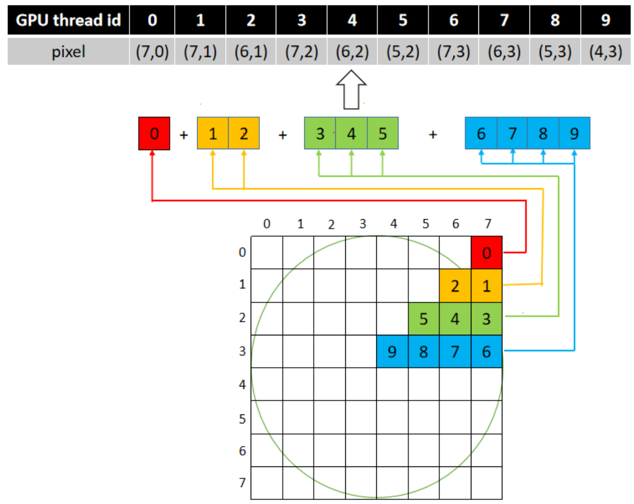


FIGURE 3. Mapping between (x, y) and GPU threads id.

and

$$\sin(2\pi nr^2 + l\theta) = \sin(2\pi nr^2)\cos(l\theta) + \cos(2\pi nr^2)\sin(l\theta). \tag{38}$$

Finally, the GPCET is given by

$$\begin{aligned} GPUM_{nl} = \frac{1}{\pi} \iint_D w(x, y) & \times (\cos(2\pi n(x^2 + y^2))G_l(x, y) \\ & - \sin(2\pi n(x^2 + y^2))H_l(x, y)) \\ & - i(\sin(2\pi n(x^2 + y^2))G_l(x, y) \\ & + \cos(2\pi n(x^2 + y^2))H_l(x, y))\, dx\, dy. \end{aligned} \tag{39}$$

$$G_l(x, y) = \begin{cases} (f(x, y) + f(y, x) + f(-y, x) + f(-x, y) \\ +f(-x, -y) + f(-y, -x) + f(y, -x) + f(x, -y))cos(l\theta) & if\ mod(l, 4) = 0 \\ (f(x, y) - f(-x, y) - f(-x, -y) + f(x, -y))cos(l\theta) \\ +(f(y, x) - f(-y, x) - f(-y, -x) + f(y, -x))sin(l\theta) & if\ mod(l, 4) = 1 \\ (f(x, y) - f(y, x) - f(-y, x) + f(-x, y) \\ +f(-x, -y) - f(-y, -x) - f(y, -x) + f(x, -y))cos(l\theta) & if\ mod(l, 4) = 2 \\ (f(x, y) - f(-x, y) - f(-x, -y) + f(x, -y))cos(l\theta) \\ -(f(y, x) - f(-y, x) - f(-y, -x) + f(y, -x))sin(l\theta) & if\ mod(l, 4) = 3, \end{cases} \tag{32}$$

$$H_l(x, y) = \begin{cases} (f(x, y) - f(y, x) + f(-y, x) - f(-x, y) \\ +f(-x, -y) - f(-y, -x) + f(y, -x) - f(x, -y))sin(l\theta) & if\ mod(l, 4) = 0 \\ (f(x, y) + f(-x, y) - f(-x, -y) - f(x, -y))sin(l\theta) \\ +(f(y, x) + f(-y, x) - f(-y, -x) - f(y, -x))cos(l\theta) & if\ mod(l, 4) = 1 \\ (f(x, y) + f(y, x) - f(-y, x) - f(-x, y) \\ +f(-x, -y) + f(-y, -x) - f(y, -x) - f(x, -y))sin(l\theta) & if\ mod(l, 4) = 2 \\ (f(x, y) + f(-x, y) - f(-x, -y) - f(x, -y))sin(l\theta) \\ -(f(y, x) + f(-y, x) - f(-y, -x) - f(y, -x))cos(l\theta) & if\ mod(l, 4) = 3, \end{cases} \tag{33}$$

TABLE 2. Running time of GPCT with different *unrollNum* and *blockDim.x*.

$blockDim.x$ \ $unrollNum$	1	2	4	8	16	32	64	128	256
1	22.362	19.364	18.034	17.405	17.079	17.099	17.452	18.269	19.922
2	11.835	10.229	9.467	9.158	9.010	9.125	9.555	10.479	10.772
4	6.205	5.440	5.101	4.954	4.963	5.152	5.776	6.514	9.786
8	3.539	3.099	2.933	2.902	3.025	3.953	4.852	6.187	9.715
16	2.255	2.108	2.134	2.213	3.0670	4.063	4.350	5.930	9.723
32	1.973	1.927	1.940	2.470	3.697	3.792	4.396	6.007	9.865
64	1.952	1.921	1.977	3.538	3.698	3.896	4.531	5.644	9.518
128	1.980	1.903	1.950	3.685	3.889	3.915	3.855	5.188	9.428
256	2.268	1.992	1.970	3.927	3.835	3.257	3.171	5.257	9.773

TABLE 3. Running time of GPST with different *unrollNum* and *blockDim.x*.

$blockDim.x$ \ $unrollNum$	1	2	4	8	16	32	64	128	256
1	22.035	19.032	17.757	17.170	16.835	16.846	17.222	18.053	19.693
2	11.566	9.980	9.243	8.916	8.789	8.927	9.326	10.274	10.518
4	6.036	5.270	4.905	4.789	4.789	4.989	5.574	6.220	9.554
8	3.416	2.967	2.789	2.754	2.870	3.717	4.514	5.838	9.501
16	2.104	1.981	1.997	2.059	2.798	3.719	3.969	5.620	9.519
32	1.839	1.786	1.786	2.262	3.337	3.389	3.973	5.631	9.654
64	1.830	1.764	1.831	3.213	3.337	3.481	4.078	5.364	9.372
128	1.805	1.761	1.810	3.312	3.495	3.486	3.528	5.041	9.236
256	2.075	1.834	1.839	3.532	3.4296	2.988	3.009	5.097	9.561

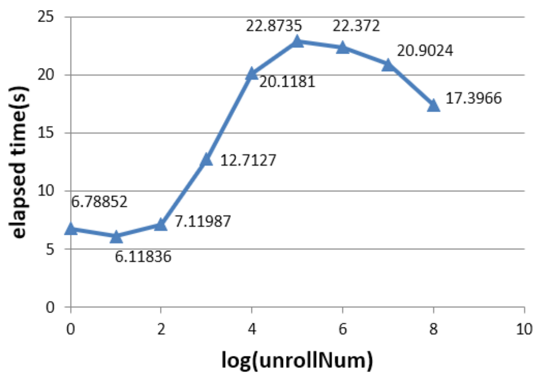


FIGURE 4. Running time of GPCT with different *unrollNum* on synthetic images.

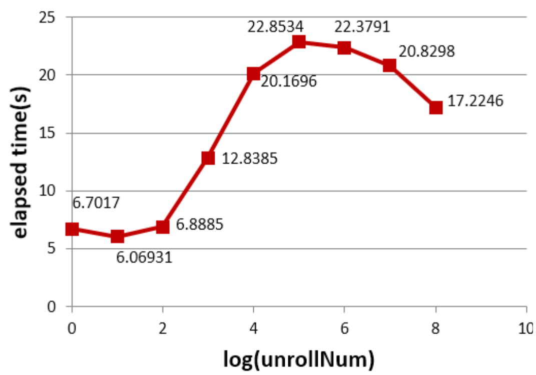


FIGURE 5. Running time of GPST with different *unrollNum* on synthetic images.

By using Eqs. (30), (36) and (39), a group of symmetric pixels can be handled by a GPU thread. In this case, Eqs. (21) and (22) should be reevaluated. As shown in FIGURE 3, we select one pixel from row 0, two pixels from row 1 and so on, then concatenate them for GPU threads. We have:

$$\frac{y \times (y+1)}{2} + (N - 1 - x) = id, \tag{40}$$

where $0 \leq x \leq y < n$. GPU thread id is in following range

$$\frac{y \times (y+1)}{2} < id \leq \frac{(y+1) \times ((y+1)+1)}{2}, \tag{41}$$

y can be deduced from id as

$$y = \left\lfloor \frac{-1 + \sqrt{1 + 8 \times id}}{2} \right\rfloor, \tag{42}$$

TABLE 4. Running time of GPCET with different *unrollNum* and *blockDim.x*.

$blockDim.x$ \ $unrollNum$	1	2	4	8	16	32	64	128	256
1	23.388	20.378	19.090	18.492	18.167	18.197	18.625	19.518	21.263
2	12.309	10.689	9.9340	9.6051	9.4964	9.6478	10.089	11.097	11.348
4	6.353	5.606	5.261	5.145	5.154	5.374	5.991	6.610	10.385
8	3.576	3.144	2.981	2.959	3.058	3.930	4.669	6.223	10.315
16	2.226	2.063	2.057	2.156	2.964	3.830	4.166	6.059	10.317
32	1.888	1.844	1.858	2.393	3.463	3.558	4.177	6.079	10.339
64	1.888	1.821	1.902	3.321	3.471	3.634	4.261	5.850	10.378
128	1.881	1.820	1.879	3.436	3.619	3.669	3.775	5.541	10.455
256	2.176	1.900	1.911	3.649	3.576	3.179	3.272	5.578	10.523

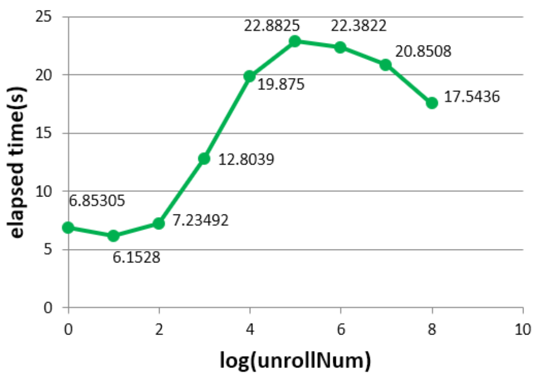


FIGURE 6. Running time of GPCET with different *unrollNum* on synthetic images.

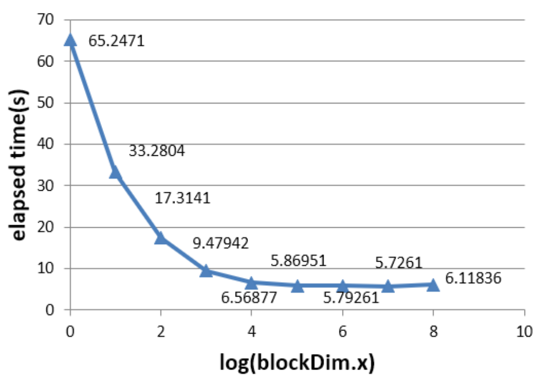


FIGURE 7. Running time of GPCT with different *blockDim.x* on synthetic images.

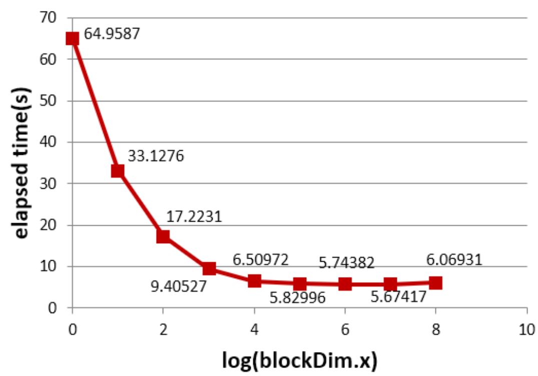


FIGURE 8. Running time of GPST with different *blockDim.x* on synthetic images.

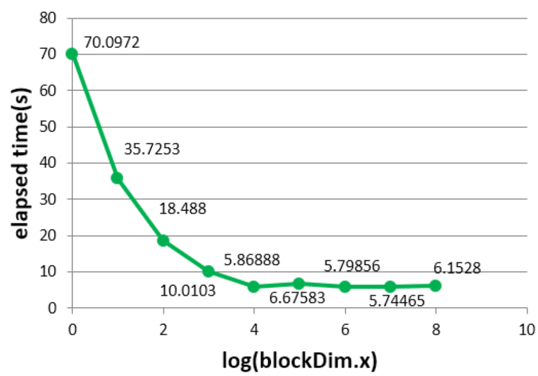


FIGURE 9. Running time of GPCET with different *blockDim.x* on synthetic images.

x can be calculated as

$$x = N - 1 - \left(id - \frac{y \times (y+1)}{2}\right). \tag{43}$$

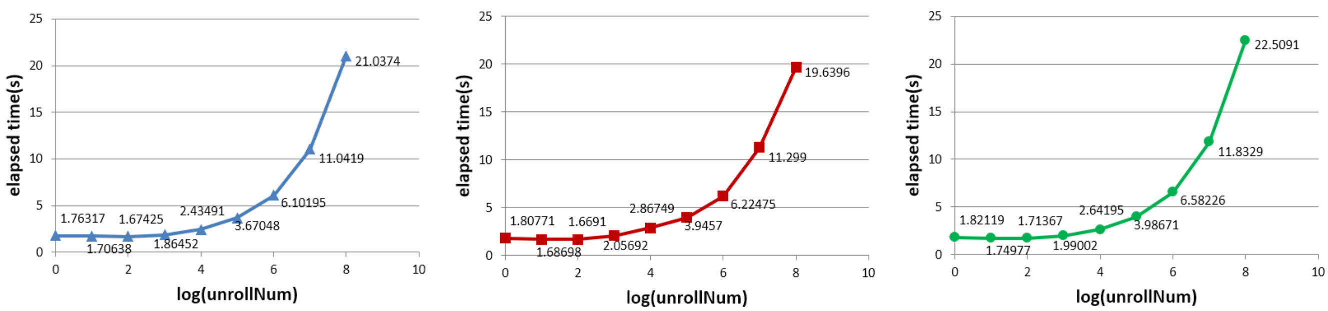
D. UNROLL OPERATION

Unroll operation is an important technique that optimizes GPU execution speed by reducing branch penalties and hiding latencies including the delay from reading data [44]. In proposed method, unroll operation is to calculate more than one group of pixels in a thread. Let *unrollNum* be the number of groups calculated in a thread. To choose an optimal *unrollNum* depends on algorithm complexity and GPU memory limitation.

IV. EXPERIMENTAL RESULTS

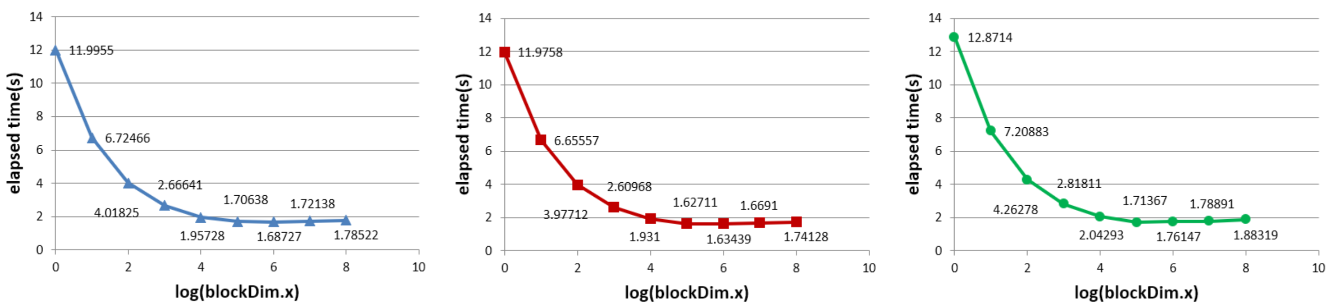
Images with different resolution and content are tested to illustrate the feasibility and efficiency of proposed GPHTs.

FIGURE 10. Real images in ITS applications.



(a) GPCT (b) GPST (c) GPCET

FIGURE 11. Running time of GPHTs with different *unrollNum* on real images.



(a) GPCT (b) GPST (c) GPCET

FIGURE 12. Running time of GPHTs with different *blockDim.x* on real images.

TABLE 5. Running time of PHTs and GPHTs on synthetic images.

Resolution	Transform	CPU(s)	GPU(s)	CPU/GPU
256 × 256	PCT	122.71	0.581	211.30
256 × 256	PST	122.78	0.570	215.33
256 × 256	PCET	88.15	0.573	153.72
512 × 512	PCT	577.29	0.688	838.51
512 × 512	PST	553.15	0.689	801.77
512 × 512	PCET	473.40	0.735	644.34
1024 × 1024	PCT	2640.70	1.903	1387.40
1024 × 1024	PST	2429.90	1.761	1493.10
1024 × 1024	PCET	2113.40	1.820	1161.20
2048 × 2048	PCT	10808.0	5.726	1887.50
2048 × 2048	PST	10187.0	5.674	1795.40
2048 × 2048	PCET	8772.40	5.745	1527.10

Windows 10 and Visual Studio 2015 are used to perform experiments, CPU (Intel Core i3-8100) has 4 cores with 3.6GHz frequency, GPU (Nvidia GeForce GTX 1060) has 1280 cores with 1594MHz frequency and graphic memory is 6GB, CUDA version is 9.0.176.

A. SYNTHETIC IMAGES

Synthetic images are used. They are generated by following formula:

$$f(i,j) = rand(1, 255), \quad 0 \le i < N,\ 0 \le j < N, \quad (44)$$

where $rand(\cdot)$ is a function to randomly generate a integer. 1,000 synthetic images are generated. In this experiment, n ranges from 11 to 16 and l ranges from 11 to 15.

With different *unrollNum* and *blockDim.x*, the running time of GPCT, GPST and GPCET are different as shown in Tables 2, 3 and 4 respectively.

unrollNum is defined as a power of two, like 2, 4, 8. For 1000 images with 2048 × 2048 resolution, FIGUREs 4, 5 and 6 show running time curve of GPCT, GPST and GPCET for different *unrollNum*. When log(*unrollNum*) is 1, GPCT, GPST and GPCET achieve the best performance.

We evaluate the impact of *blockDim.x*. *blockDim.x* is defined as a power of two, like 2, 4, 8. FIGUREs 7, 8 and 9 show running time curve of GPCT, GPST and GPCET for 1000 synthetic images. As for GPCT when log(*blockDim.x*) is 7, proposed method is the fastest and is about 11.39 times comparing to log(*blockDim.x*) is 0. Similarly, GPST and GPCET achieve the best performance when log(*blockDim.x*) is 7.

With optimal *unrollNum* and *blockDim.x*, we evaluate GPHTs against PHTs as shown in Table 5. While image resolution increasing, the proposed GPHTs outperform obviously. In our experiment for images with 2048 × 2048 resolution, GPCT runs 1887.5 times faster than PCT on CPU. GPST can achieve 1795.4 times faster than PST on CPU. GPCET is 1527.1 times faster than PCET on CPU.

TABLE 6. Running time of PHTs and GPHTs on real images.

Resolution	Transform	CPU(s)	GPU(s)	CPU/GPU
512 × 512	PCT	1610.4	1.6742	961.89
512 × 512	PST	1628.0	1.6691	975.38
512 × 512	PCET	1353.8	1.7136	790.03

B. REAL IMAGES

ITS real images are shown in FIGURE 10. 128 image patches with 512 × 512 resolution are selected. In this experiment, n ranges from 1 to 25 and l ranges from 1 to 25.

As shown in FIGURE 11 when log(*unrollNum*) is 2, GPCT, GPST and GPCET achieve the best performance. We also evaluate optimal *blockDim.x* as shown in FIGURE 12. When log(*blockDim.x*) is 7, GPCT, GPST and GPCET achieve the best performance.

With optimal *unrollNum* and *blockDim.x*, Table 6 shows the running time comparison of GPHTs and PHTs. In our experiment for real images with 512 × 512 resolution, GPCT runs 961.8 times faster than PCT on CPU. GPST can achieve 975.3 times faster than PST on CPU. GPCET is 790 times faster than PCET on CPU.

V. CONCLUSION

In this paper, we propose GPU based PHT. By using the symmetric properties and mathematical properties of trigonometric functions, parallel GPU threads can manipulate pixels simultaneously. Formulas between GPU thread *id* and image pixel (x, y) are deduced. For real time systems and large multimedia databases, proposed method can fully unleash GPU parallel computational capability. Comprehensive experiments are also given to illustrate the effectiveness of proposed method. Wide range of emerging applications that using PHTs will be inspired from this study.

REFERENCES

[1] W.-L. Zheng, K. Gao, G. Li, W. Liu, C. Liu, J.-Q. Liu, G. Wang, and B.-L. Lu, "Vigilance estimation using a wearable EOG device in real driving environment," *IEEE Trans. Intell. Transp. Syst.*, vol. 21, no. 1, pp. 170–184, Jan. 2020.

[2] T. Deng, K. Yang, Y. Li, and H. Yan, "Where does the driver look? Top-Down-Based saliency detection in a traffic driving environment," *IEEE Trans. Intell. Transp. Syst.*, vol. 17, no. 7, pp. 2051–2062, Jul. 2016.

[3] Q. Zhang and S.-I. Kamata, "Improved color barycenter model and its separation for road sign detection," *IEICE Trans. Inf. Syst.*, vol. E96.D, no. 12, pp. 2839–2849, 2013.

[4] Q. Zhang and S.-I. Kamata, "A novel color descriptor for road-sign detection," *IEICE Trans. Fundam. Electron., Commun. Comput. Sci.*, vol. E96.A, no. 5, pp. 971–979, 2013.

[5] N. K. Ragesh and R. Rajesh, "Pedestrian detection in automotive safety: Understanding state-of-the-art," *IEEE Access*, vol. 7, pp. 47864–47890, 2019.

[6] F. Bu, T. Le, X. Du, R. Vasudevan, and M. Johnson-Roberson, "Pedestrian planar LiDAR pose (PPLP) network for oriented pedestrian detection based on planar LiDAR and monocular images," *IEEE Robot. Autom. Lett.*, vol. 5, no. 2, pp. 1626–1633, Apr. 2020.

[7] Z. Ren, Q. Zhang, X. Gao, P. Hao, and J. Cheng, "Multi-modality learning for human action recognition," *Multimedia Tools Appl.*, pp. 1–19, Mar. 2020, doi: 10.1007/S11042-019-08576-Z.

[8] Y. Tian, Q. Zhang, Z. Ren, F. Wu, P. Hao, and J. Hu, "Multi-scale dilated convolution network based depth estimation in intelligent transportation systems," *IEEE Access*, vol. 7, pp. 185179–185188, 2019.

[9] Y. Tian, Y. Du, Q. Zhang, J. Cheng, and Z. Yang, "Depth estimation for advancing intelligent transport systems based on self-improving pyramid stereo network," *IET Intell. Transp. Syst.*, vol. 14, no. 5, pp. 338–345, May 2020.

[10] Z. Ning, P. Dong, X. Wang, X. Hu, L. Guo, B. Hu, Y. Guo, T. Qiu, and R. Y. K. Kwok, "Mobile edge computing enabled 5G health monitoring for Internet of medical things: A decentralized game theoretic approach," *IEEE J. Sel. Areas Commun.*, to be published.

[11] Z. Ning, R. Y. K. Kwok, K. Zhang, X. Wang, M. S. Obaidat, L. Guo, X. Hu, B. Hu, Y. Guo, and B. Sadoun, "Joint computing and caching in 5G-envisioned Internet of vehicles: A deep reinforcement learning-based traffic control system," *IEEE Trans. Intell. Transp. Syst.*, early access, Feb. 2, 2020, doi: 10.1109/TITS.2020.2970276.

[12] Z. Ning, Y. Li, P. Dong, X. Wang, M. S. Obaidat, X. Hu, L. Guo, Y. Guo, J. Huang, and B. Hu, "When deep reinforcement learning meets 5G-enabled vehicular networks: A distributed offloading framework for traffic big data," *IEEE Trans. Ind. Informat.*, vol. 16, no. 2, pp. 1352–1361, Feb. 2020.

[13] X. Hu, J. Cheng, M. Zhou, B. Hu, X. Jiang, Y. Guo, K. Bai, and F. Wang, "Emotion-aware cognitive system in multi-channel cognitive radio ad hoc networks," *IEEE Commun. Mag.*, vol. 56, no. 4, pp. 180–187, Apr. 2018.

[14] J. Zhang, X. Hu, Z. Ning, E. C.-H. Ngai, L. Zhou, J. Wei, J. Cheng, and B. Hu, "Energy-latency tradeoff for energy-aware offloading in mobile edge computing networks," *IEEE Internet Things J.*, vol. 5, no. 4, pp. 2633–2645, Aug. 2018.

[15] P.-T. Yap, X. Jiang, and A. C. Kot, "Two-dimensional polar harmonic transforms for invariant image representation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 32, no. 7, pp. 1259–1270, Jul. 2010.

[16] A. Farnoosh and N.-G. Ali, "Accurate body-part reconstruction from a single depth image," *Multimedia Syst.*, vol. 25, no. 3, pp. 165–176, Jun. 2019.

[17] Z. Al-asady and A. Al-amery, "Human action recognition using a corners and blob detector with different classification methods," *Int. Conf. Sustain. Eng. Techn.*, vol. 518, Art. no. 052008, May 2019.

[18] C. Lin, W. Lu, X. Huang, K. Liu, W. Sun, and H. Lin, "Region duplication detection based on hybrid feature and evaluative clustering," *Multimedia Tools Appl.*, vol. 78, no. 15, pp. 20739–20763, Aug. 2019, doi: 10.1007/s11042-019-7342-9.

[19] Y. Liu, S. Zhang, G. Li, H. Wang, and J. Yang, "Accurate quaternion radial harmonic Fourier moments for color image reconstruction and object recognition," *Pattern Anal. Appl.*, pp. 1–17, Apr. 2020, doi: 10.1007/s10044-020-00877-6.

[20] L. Li, J. Zhang, and A. Abraham, "Image watermarking based on invariant representation of polar sine transform," *IEICE Trans. Fundamentals Electron., Commun. Comput. Sci.*, vols. E94–A, no. 10, pp. 2048–2052, 2011.

[21] L. Li, S. Li, A. Abraham, and J.-S. Pan, "Geometrically invariant image watermarking using polar harmonic transforms," *Inf. Sci.*, vol. 199, pp. 1–19, Sep. 2012.

[22] C. Singh and S. K. Ranade, "Rotation invariant moments and transforms for geometrically invariant image watermarking," *J. Electron. Imag.*, vol. 22, no. 1, Mar. 2013, Art. no. 013034.

[23] M. Qi, B.-Z. Li, and H. Sun, "Image watermarking via fractional polar harmonic transforms," *J. Electron. Imag.*, vol. 24, no. 1, Jan. 2015, Art. no. 013004.

[24] X.-Y. Wang, Y.-N. Liu, S. Li, H.-Y. Yang, and P.-P. Niu, "Robust image watermarking approach using polar harmonic transforms based geometric correction," *Neurocomputing*, vol. 174, pp. 627–642, Jan. 2016.

[25] M. Liu and P.-T. Yap, "Invariant representation of orientation fields for fingerprint indexing," *Pattern Recognit.*, vol. 45, no. 7, pp. 2532–2542, Jul. 2012.

[26] Y. Li, "Image copy-move forgery detection based on polar cosine transform and approximate nearest neighbor searching," *Forensic Sci. Int.*, vol. 224, nos. 1–3, pp. 59–67, Jan. 2013.

[27] L. Li, S. Li, H. Zhu, and X. Wu, "Detecting copy-move forgery under affine transforms for image forensics," *Comput. Electr. Eng.*, vol. 40, no. 6, pp. 1951–1962, Aug. 2014.

[28] D. Cozzolino, G. Poggi, and L. Verdoliva, "Efficient dense-field copy–move forgery detection," *IEEE Trans. Inf. Forensics Security*, vol. 10, no. 11, pp. 2284–2297, Nov. 2015.

[29] K. M. Hosny, H. M. Hamza, and N. A. Lashin, "Copy-move forgery detection of duplicated objects using accurate PCET moments and morphological operators," *Imag. Sci. J.*, vol. 66, no. 6, pp. 330–345, Aug. 2018.

[30] Y. Nan Li, "Quaternion polar harmonic transforms for color images," *IEEE Signal Process. Lett.*, vol. 20, no. 8, pp. 803–806, Aug. 2013.

[31] S. P. Singh and S. Urooj, "Three types of moment invariants for color object recognition based on radon and polar harmonic transform in Cℓ (0, 2) space," *Arabian J. Sci. Eng.*, vol. 41, no. 8, pp. 3051–3060, Aug. 2016.

[32] S. P. Singh, S. Urooj, and A. Lay-Ekuakille, "Breast cancer detection using PCPCET and ADEWNN: A geometric invariant approach to medical X-ray image sensors," *IEEE Sensors J.*, vol. 16, no. 12, pp. 4847–4855, Jun. 2016.

[33] J. Wang, G. Wang, M. Li, and W. Du, "Hand vein recognition based on PCET," *Optik*, vol. 127, no. 19, pp. 7663–7669, Oct. 2016.

[34] J. Wu, S. Qiu, Y. Kong, Y. Chen, L. Senhadji, and H. Shu, "MomentsNet: A simple learning-free method for binary image recognition," in *Proc. IEEE Int. Conf. Image Process. (ICIP)*, Sep. 2017, pp. 2667–2671.

[35] Z. Yang, G. Chen, and D. Shen, "Robust construction of diffusion MRI atlases with correction for inter-subject fiber dispersion," in *Computational Diffusion MRI*. VA, USA: Springer, 2017, pp. 113–121.

[36] Z. Yang, G. Chen, D. Shen, and P.-T. Yap, "Robust fusion of diffusion MRI data for template construction," *Sci. Rep.*, vol. 7, no. 1, Dec. 2017, Art. no. 12950.

[37] H. Chen, Y. Wo, and G. Han, "Multi-granularity geometrically robust video hashing for tampering detection," *Multimedia Tools Appl.*, vol. 77, no. 5, pp. 5303–5321, Mar. 2018.

[38] W. Tang, Y. Wo, and G. Han, "Geometrically robust video hashing based on ST-PCT for video copy detection," *Multimedia Tools Appl.*, vol. 78, no. 15, pp. 21999–22022, Aug. 2019, doi: 10.1007/s11042-019-7513-8.

[39] G. Kuppudurai, K.-Y. Hwang, H.-G. Park, and Y. Kim, "Localization of airborne platform using digital elevation model with adaptive weighting inspired by information theory," *IEEE Sensors J.*, vol. 18, no. 18, pp. 7585–7592, Sep. 2018.

[40] Y.-N. Liu, S.-S. Zhang, Y. Sang, and S.-M. Wang, "Improving image retrieval by integrating shape and texture features," *Multimedia Tools Appl.*, vol. 78, no. 2, pp. 2525–2550, Jan. 2019.

[41] C. Song, J. Lei, J. Guo, and Y. Li, "Block-based two-dimensional wavelet transform running on graphics processing unit," *IET Comput. Digit. Techn.*, vol. 8, no. 5, pp. 229–236, Sep. 2014.

[42] H. Bahri, F. Sayadi, R. Khemiri, M. Chouchene, and M. Atri, "Image feature extraction algorithm based on CUDA architecture: Case study GFD and GCFD," *IET Comput. Digit. Techn.*, vol. 11, no. 4, pp. 125–132, Jul. 2017.

[43] J. Sanders and E. Kandrot, *CUDA by Example: An Introduction to General-Purpose GPU Programming*. Reading, MA, USA: Addison-Wesley, 2010.

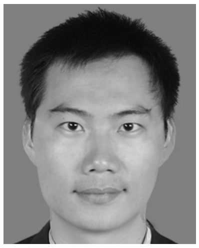
[44] J. Cheng, M. Grossman, and T. McKercher, *Professional CUDA C Programming*. Birmingham, U.K.: Wrox Press, 2014.

[45] C. S. Burrus and T. W. Parks, *DFT/FFT and Convolution Algorithms and Implementation*. Hoboken, NJ, USA: Wiley, 1985.

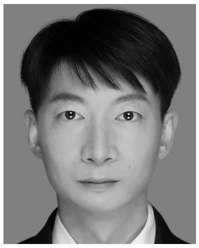
ZHUO YANG received the B.S. and M.S. degrees from the Beijing Institute of Technology, in 2005 and 2008, respectively, and the Ph.D. degree from Waseda University, in 2012. He is currently an Assistant Professor with the School of Computers, Guangdong University of Technology. His main research interests include image processing, computer vision, and VR/AR.

MINGKAI TANG is currently pursuing the bachelor's degree with the Guangdong University of Technology. His main research interests include image processing and computer vision.

ZHUOZHANG LI is currently pursuing the bachelor's degree with the Guangdong University of Technology. His main research interests include image processing and artificial intelligence.



ZILIANG REN received the Ph.D. degree from the South China University of Technology, Guangzhou, China, in 2017. He is currently holding a postdoctoral position and an Assistant Researcher with the Guangdong Provincial Key Laboratory of Robotics and Intelligent System, Shenzhen Institutes of Advanced Technology, Chinese Academy of Sciences, Shenzhen, China. His current research interests include computer vision and machine learning.



QIESHI ZHANG (Member, IEEE) received the Ph.D. degree from Waseda University, Japan, in 2014. From 2012 to 2016, he was a Research Assistant and a Research Associate with the Information, Production, and Systems Research Center, Waseda University. He is currently an Associate Professor with the Guangdong Provincial Key Laboratory of Robotics and Intelligent System, Shenzhen Institutes of Advanced Technology, Chinese Academy of Sciences, China. He has authored or coauthored over 70 scientific articles in international journals and conferences. His current research interests include computer vision, autonomous driving, and intelligent robots. He serves as the technical/program committee member for over 50 conferences and over 100 times. From 2010 to 2012, he was a Research Fellow with the Japan Society for the Promotion of Science (JSPS), Japan.