IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received September 13, 2018, accepted November 1, 2018, date of publication December 12, 2018,
date of current version December 31, 2018.

Digital Object Identifier 10.1109/ACCESS.2018.2879919

Parallel Multi-Core CPU and GPU for Fast and Robust Medical Image Watermarking

KHALID M. HOSNY[1], (Member, IEEE), MOHAMED M. DARWISH[2], KENLI LI[3], AND AHMAD SALAH[3,4]

[1]Department of Information Technology, Faculty of Computers and Informatics, Zagazig University, Zagazig 44519, Egypt
[2]Department of Mathematics, Faculty of Science, Assiut University, Assiut 71516, Egypt
[3]College of Computer Science and Electronic Engineering, Hunan University, Hunan 410082, China
[4]Department of Computer Sciences, Faculty of Computers and Informatics, Zagazig University, Zagazig 44519, Egypt

Corresponding author: Khalid M. Hosny (k_hosny@yahoo.com)
This work was supported in part by the National Key R&D Program of China under Grant SQ2018YFB020061, in part by the Program of China under Grant 2016YFB0201402, in part by The International (Regional) Cooperation and Exchange Program of National Natural Science Foundation of China under Grant 61860206011, and in part by the National Natural Science Foundation of China under Grant 61702170, Grant 61602350, Grant 61602170, and Grant 61750110531.

ABSTRACT Securing medical images are a very essential process in medical image authentication. Medical image watermarking is a very popular tool to achieve this goal. In this paper, an extremely fast, highly accurate, and robust algorithm is proposed for watermarking both gray-level and color medical images. In the proposed method, a scrambled binary watermark is embedded in the host medical image. Simplified exact kernels are used to compute the moments of the polar complex exponential transform (PCET) for the host gray-level images and the moments of the quaternion PCET for the host color images without approximation errors. The stability of the computed moments enables us to use higher order moments in a perfect reconstruction of the watermarked medical images. The accurate moment invariant to rotation, scaling, and translation ensures the robustness of the proposed watermarking algorithm against geometric attacks. Performed experiments clearly show very high visual imperceptibility and robustness to different levels of geometric distortions and common signal processing attacks. The implementation of parallel multi-core CPU and GPU result in a tremendous reduction of the overall watermarking times. For a color image of size 256 × 256, the watermarking time is accelerated by 20× and 11× using a GPU and a CPU with 16 cores, respectively.

INDEX TERMS Medical image watermarking, telemedicine, polar harmonic transforms, geometric attacks, parallel architecture, multi-core CPUs, GPU.

I. INTRODUCTION

Around the world, medical imaging devices produces several millions of medical images every day. These medical images required several megabytes of disk space. These medical images are stored in the medical files of patients for at least two decades. Medical files of patients are transmitted through the health networks for medical diagnosis or legal purposes [1]. Medical images by nature are sensitive to changes where any change in its contents may results in an inaccurate medical diagnosis. Securing the medical images is a big challenge where the medical images must maintain their original contents at the time of reconstruction [2].

Medical images watermarking is a famous approach used in securing medical images during their storage and transmission through different healthcare networks [3]. The medical digital watermarking is the process of hiding a digital watermark in the original medical image where the hidden digital watermark plays an essential role in protecting the contents of the medical images. The digital watermark is embedded and then the watermarked image is stored. To ensure the originality of the stored medical images, any change in the content of these medical images could be easily noticed by detecting the watermark. In image medical transmission, the embedded watermark should be identical on both sides, the sender and receiver [4].

Medical images are transmitted among the patient, primary physician, and the referred physicians. Medical information suffers from the risk of security issues including confidentiality, reliability, and authenticity [5]. Therefore, medical image watermarking is widely used as a

security means to prevent any malicious or inadvertent changes [6], [7].

The objectives of medical image watermarking (MIW) can be realized in two-fold. The first is to control integrity and authentication and the second is to hide important data, i.e. the electronic patient record (EPR) information [8].

Qasim *et al.* [9], defined the digital watermarking as a promising approach for ensuring the authenticity and integrity of medical images. Authenticity refers to the ability to identify the information origin and prove that the data relates to the right patient. Integrity means the capacity to ensure that the information has not been altered without authorization.

Generally, the watermarking process facing different problems such as signal processing and geometric attacks. Common signal processing attacks such as different filters, different kind of additive noise with different levels of contamination, and image compression. Geometric attacks or distortions include rotation, scaling, and translation (RST).

During the recent years, remarkable algorithms are proposed for medical image watermarking [10]–[15]. In these algorithms, the authors paid their attention to handle the common signal processing attacks and ignore the geometric distortions. However, a successful and reliable medical watermarking algorithm must be robust to individual or multiple geometric as well as common signal processing attacks. The combination of different attacks is a big challenging for medical image watermarking systems due to the sensitive nature of the medical images.

Orthogonal moments are widely used in image reconstruction and recognition with a minimum amount of information redundancy [16]. Invariance to rotation, scaling and translations could be achieved through a set of mathematical procedures. The orthogonal moment invariants are very useful descriptors for gray and color images. Based on this attractive characteristic, a number of geometrically moment-based image watermarking techniques were developed for gray-level images [17]–[25]. Since color images have more information than their corresponding gray-level images, watermarking of color images received more attention during the last decade where the embedded digital watermark may be a gray or color image. Recently, quaternion-moment based watermarking methods of color images has been started as a promising area of research [26]–[30].

Recently, scientists utilized the graphic processing unit (GPU) to accelerate the watermarking process. Fang *et al.* [31] presented techniques for efficient Discrete Cosine Transform (DCT) and an improved DCT (IDCT) implementation on GPU for gray-level images. Islam *et al.* [32] used (8, 4) Hamming code-based error correction in gray-level medical images watermarking. They implemented the Hamming code on a graphics processing unit (GPU) to accelerate and meet the real-time requirements. Khor *et al.* [33] presented a parallel watermarking algorithm of gray-level ultrasound medical images in multi-cores environment. Abadi and Chalechale [34] presented a parallelization of a color image DCT-based watermarking algorithm using a CUDA-based approach. Fan *et al.* [35] presented a DCT-based watermarking method for color images where they added invisible watermark into one channel of color RGB images in the frequency domain. They used the GPU to accelerate the embedding process.

The authors of the works [31]–[34] used approximate DCT in gray-level and color images. The utilization of the inaccurate DCT coefficients results in accumulation of approximation errors which degraded the quality of the watermarked images. In addition to this disadvantage, the DCT-based watermarked algorithms are time-consuming. Fan *et al.* [35] applied a block-based method to accelerate the watermarking process where the input images are divided into non-overlapped blocks which produced what is called blocking artifacts [36]. These artifacts reduce the quality of the watermarked images, especially the medical images. These limitations motivate the authors to present the fast and robust medical image watermarking algorithm.

In this paper, a geometrically invariant robust watermarking algorithm is proposed where the orthogonal Polar Complex Exponential Transform (PCET) and its quaternion model (QPCET) are used in watermarking gray-level and color medical images, respectively. The highly accurate computation of both PCET and QPCET using the simplified kernel approach result in a perfect reconstruction of the watermarked medical images. Moreover, these moments are more suitable for robust medical image watermarking where the highly accurate and numerically stable PCET and QPCET moments provided us a better visual imperceptibility and higher robustness against individual and multiple geometric distortions and common signal processing attacks. Tremendous reduction in the computational cost through the parallel implementation of the proposed watermarking algorithm on multi-core CPUs and GPUs makes it fast and suitable for real-time medical image authentication.

Numerical simulation through a set of conducted experiments is performed. The obtained results show a better performance in terms of robustness to all kind of attacks and visual imperceptibility.

The rest of this paper is organized as follows. Section II presents preliminaries about the PCET for gray-level images and QPCET for color images. An introduction to parallel architectures is presented in Section III. The proposed watermarking of medical images on multi-core CPUs and GPU are presented in Section IV. Numerical experiments for watermarking gray-level and color medical images are described in Section V. A conclusion is presented in Section VI.

II. POLAR HARMONIC TRANSFORMS

Yap *et al.* [37] defined the polar harmonic transforms for gray-level images. These three transforms are defined based on their basis functions, namely, polar complex exponential transform (PCET), polar cosine transform (PCT) and polar

sine transform (PST). The PCET moments for a gray-level image is defined as follows:

$$A_{mn} = \frac{1}{\pi}\int_0^{2\pi}\int_0^1 [H_{mn}(r,\theta)]^* f(r,\theta)\, r dr d\theta \quad (1)$$

where $f(r,\theta)$ refers to the image intensity function; the indices m and n refer to the order the repetition respectively. Their values are $|m| = |n| = 0, 1, 2, 3, \ldots\ldots\ldots\infty$; the operator $[\blacksquare]^*$ refers to the complex conjugate. The orthogonal basis function, $H_{mn}(r,\theta)$, are defined as follows:

$$H_{mn}(r,\theta) = R_m(r)\, e^{-\hat{i}n\theta} = e^{-\hat{i}2\pi mr^2} e^{-\hat{i}n\theta} \quad (2)$$

These polynomials obeyed the following orthogonality relation:

$$\int_0^{2\pi}\int_0^1 H_{mn}(r,\theta)\,[H_{st}(r,\theta)]^* r dr d\theta = \pi\delta_{ms}\delta_{nt} \quad (3)$$

where $\hat{i} = \sqrt{-1}$ and δ_{ms} is the Kronecker function.

A. QUATERNION POLAR HARMONIC TRANSFORMS

The quaternion numbers were defined by Hamilton [38] as a generalization of the complex numbers. A quaternion number, q, consists of four components, one real and three imaginary and expressed as follows:

$$q = a + b\boldsymbol{i} + c\boldsymbol{j} + d\boldsymbol{k} \quad (4)$$

where $\boldsymbol{i}, \boldsymbol{j}$, and $\boldsymbol{k}$ are imaginary units and a, b, c, and d are real numbers. The imaginary units obeying the relations:

$$\begin{aligned} &\mathbf{i^2} = \mathbf{j^2} = \mathbf{k^2} = \mathbf{ijk} = -1 \\ &\mathbf{ij} = -\mathbf{ji} = \mathbf{k}; \quad \mathbf{jk} = -\mathbf{kj} = \mathbf{i}; \ \mathbf{ki} = -\mathbf{ik} = \mathbf{j} \end{aligned} \quad (5)$$

The conjugate and the modulus of the quaternion are defined as follows:

$$\begin{aligned} q^* &= a - b\boldsymbol{i} - c\boldsymbol{j} - d\boldsymbol{k}, \\ |q| &= \sqrt{a^2 + b^2 + c^2 + d^2} \end{aligned} \quad (6)$$

If $a = 0$, the quaternion, q, is called a pure quaternion. Ell and Sangwine [39] showed that a color image, $f(x,y)$ can be separated into three channels, red, green and blue and represented by using a pure quaternion as follows:

$$f(x,y) = f_R(x,y)\,\boldsymbol{i} + f_G(x,y)\,\boldsymbol{j} + f_B(x,y)\,\boldsymbol{k} \quad (7)$$

where $f_R(x,y)$, $f_G(x,y)$, and $f_B(x,y)$ represent the red, green and blue components of the color pixel respectively.

Wang et. al. [40] defined the right-side quaternion polar complex exponential transform (QPCET) for the RGB color image, $f(r,\theta)$, in polar coordinates as follows:

$$\begin{aligned} M^R_{mn} &= \frac{1}{\pi}\int_0^{2\pi}\int_0^1 f(r,\theta)\,[E_{mn}(r,\theta)]^* r dr d\theta \\ &= \frac{1}{\pi}\int_0^{2\pi}\int_0^1 f(r,\theta)\exp\left(-\mu 2\pi mr^2\right)\exp(-\mu n\theta) \\ &\quad \times r dr d\theta \end{aligned} \quad (8)$$

where $\mu = (\boldsymbol{i}+\boldsymbol{j}+\boldsymbol{k})/\sqrt{3}$ is called a unit pure quaternion. The quaternion basis functions, $E_{mn}(r,\theta)$, are defined as follows:

$$E_{mn}(r,\theta) = e^{\mu 2\pi mr^2} e^{\mu n\theta} \quad (9)$$

Based on Equation (3), the PCET moments are used to reconstruct gray-level images as follows:

$$f^{Recon.}(r,\theta) = \sum_{m=-\infty}^{\infty}\sum_{n=-\infty}^{\infty} A_{mn} H_{mn}(r,\theta), \quad (10)$$

Since the computing platforms handling only definite values and cannot accept the infinity, the computations are limited to specific maximum numbers m_Max and n_Max. Equation (10) is rewritten as follows:

$$f^{Recon.}(r,\theta) \approx \sum_{m=-\mathrm{m\_Max}}^{\mathrm{m\_Max}}\sum_{n=-\mathrm{n\_Max}}^{\mathrm{n\_Max}} A_{mn} H_{mn}(r,\theta), \quad (11)$$

The PCET moments with negative order and repetitions such as $A_{-m,-n}$ are directly computed using the following equation:

$$A_{-m,-n} = (A_{m,n})^* \quad (12)$$

Similarly, the color image, $f(r,\theta)$, could be reconstructed by using the QPCET moments for $m \le m\_Max, n \le n\_Max$ as follows:

$$\hat{f}(r,\theta) \approx \sum_{m=-m\_Max}^{m\_Max}\sum_{n=-n\_Max}^{n\_Max} M^R_{mn} e^{\mu 2\pi mr^2} e^{\mu n\theta} \quad (13)$$

III. INTRODUCTION TO PARALLEL ARCHITECTURES

A. MULTI-CORE CPUs

In multi-core CPU, the number of available resources is limited; it is the number of available cores. CPU cores are computationally powerful tool designed to handle sequential codes efficiently. The recent advances in CPU architectures enable multiprocessing through multi-core CPU. The entire cores of CPU can access the main memory. In addition, concurrent programming for multi-core CPUs are affordable due to the existing of different parallel programming libraries such as OpenMP [41].

B. MANY-CORE GPUs

Utilization of GPU for speeding up the sequential code is successfully reported in different directions [42]–[46]. In several works, the parallel code execution-time is less than the sequential counterpart by two or three orders of magnitude. GPU has hundreds or thousands of less powerful cores, in comparison to CPU cores. GPU is designed to provide a high level of parallelism through several physical streaming multiprocessors (SMs), where each SM consists of tens or hundreds of physical cores. The cores are managed to perform

synchronized tasks at a high level. The purpose of supplying a GPU with many SMs is to guarantee a high level of concurrency where each SM has its own code control unit, registers, and shared memory. In addition, the global memory space the entire SMs can access it.

CUDA [47] and OpenCL [48] are the most famous programming frameworks for GPUs. The GPU program is organized into logical blocks; each block contains a set of logical threads. To run the program on a GPU device, the logical blocks are distributed over the physical SMs and in turn the treads are assigned to the physical cores of an SM. Each SM assigns the threads in warps of size 32 or 64 threads to the physical cores. Thus, it is preferred that the number of threads per a block is a multiple of the SM warp size. If a block has one more thread than the warp size, then there is only one running warp containing only one thread, and the other warp(s) are waiting their time slice to run. It is preferred to have more than one warp per a block to hide the memory latency.

IV. THE PROPOSED WATERMARKING ALGORITHM

A detailed description of the proposed medical image watermarking algorithm is presented in this section. The procedures of embedding, detection and extraction of the watermark are described in the first subsection. The proposed procedures for fast and highly accurate computation of PCET/QPCET moments using multi-core CPUs and GPUs parallel architectures are presented in the second subsection.

A. WATERMARK EMBEDDING

Before embedding the watermark into the original host medical image, the binary watermark image content is perturbed and becomes unreadable by using the scrambling algorithm. In order to achieve more security and make the watermark be hard to be extracted, a watermark scrambling technique is frequently used in image watermarking algorithms. Since Arnold scrambling technique [29] is intuitionistic, simple, periodic, and easy to use; the author utilized this well-known scrambling technique. Let,

$$B = \{b\,(i,j) \in \{0,1\}\,, 0 \le i < P,\ 0 \le j < Q\}$$

is the binary watermark image of size $P \times Q$. The binary watermark image B is scrambled to B_1 where the scrambled image is defined as follows:

$$B_1 = \{b_1\,(i,j) \in \{0,1\}\,, 0 \le i < P,\ 0 \le j < Q\}.$$

For more simplicity, the scrambled binary watermark image is transformed into a one-dimensional bit sequence as follows:

$$B_2 = \{b_2\,(l) = b_1\,(i,j) \in \{0,1\}\,, \quad 0 \le i < P,\ 0 \le j < Q,\ l = i \times Q + j\}.$$

After the scrambling process, the embedding process is described through the following subsections.

1) SELECTION OF THE ACCURATE MOMENTS

The robustness of the proposed watermarking algorithm is enhanced by selecting the most suitable accurate PCET/QPCET moments based on two factors:

- The first one, PCET/QPCET moments with $n = 4m, m \in Z$ $(i.e. n = 0, n = 4, n = 8, n = 12, \ldots\ldots..)$ are dropped from the selection process where are not suitable for encoding watermark bits [17].
- The second factor, only the independent PCET/QPCET moments with positive repetition $n > 0$ are used. The PCET/QPCET moments with negative repetition $n < 0$ are dependent, and then are dropped to avoid information redundancy.

Therefore, the independent and accurate final moment set Φ used for watermark embedding in the proposed scheme based on selection process could be described as follows:

$$S = \{\Phi_{mn}, n \ne 4m, m \in Z\}$$

For a watermark bit sequence of length equal to $l = P \times Q$, the performance of the watermarking algorithm could be increased by selecting form the feature vector:

$$\Phi\,(l) = \left\{\Phi_{m_1 n_1}, \Phi_{m_2 n_2}, \ldots\ldots\ldots, \Phi_{m_l n_l}\right\}$$

2) DIGITAL WATERMARK EMBEDDING

The binary watermark could be embedded by modifying the magnitudes of the selected accurate moments. For the scrambled sequence B_2, the moments, $\phi(l)$, are selected from the moments set S and then the bits of the digital watermark are embedded by modifying the magnitude of the moments by using the dither modulation function [17]:

$$\Phi'(l) = \left[\frac{\Phi(l) - d_k(b_2(l))}{\Delta}\right] * \Delta + d_k\,(b_2\,(l))$$
$$0 \le l < P \times Q, \quad d_k\,(1) = \frac{\Delta}{2} + d_k\,(0)\,,\ d_k\,(0) \in [0,1] \tag{14}$$

where $\Phi(l)$ are the unmodified moments of host image while $\Phi'(l)$ are the modified moments' and the quantized version of $\Phi(l)$. The $[\cdot]$, Δ, and $d_k(\cdot)$ refer to the rounding operator, the quantization steps and the dither function associated with the K key, respectively.

3) RECONSTRUCTION OF THE WATERMARKED IMAGE

After modifying the selected PCET and QPCET moments coefficients, the watermarked medical image, $f_w(r,\theta)$, is reconstructed using composed two components (parts). One part is the image components formed by those unchanged PCET and QPCET moments for gray-level and color medical images respectively, which is defined by using the following form [29]:

$$f_{rem}\,(r,\theta) = f\,(r,\theta) - f_M\,(r,\theta) \tag{15}$$

where the second term is the image components formed by the selected PCET and QPCET moments before they are

changed,

$$f_M(r,\theta) = \sum_{i=0}^{P\times Q-1} \Phi_{m_i n_i} H_{m_i n_i}(r,\theta) + \Phi_{m_i,-n_i} H_{m_i,-n_i}(r,\theta) \quad (16)$$

The other part is the image components formed by those modified PCET and QPCET moments

$$f_{M'}(r,\theta) = \sum_{i=0}^{P\times Q-1} \Phi'_{m_i n_i} H_{m_i n_i}(r,\theta) + \Phi'_{m_i,-n_i} H_{m_i,-n_i}(r,\theta) \quad (17)$$

Finally, the watermarked medical image $f_w(r,\theta)$ could be obtained by combining the two parts where $f_w(r,\theta)$ was formed as:

$$f_W(r,\theta) = f_{rem}(r,\theta) + f_{M'}(r,\theta) \quad (18)$$

B. WATERMARK EXTRACTION

The binary watermark information is extracted from the selected PCET/QPCET moments for watermarked image f_w. In the following subsections, we will explain the watermark extraction process at the detector's side, which is similar to the embedding process.

1) SELECTION OF ACCURATE FEATURES

The PCET/QPCET moments of the possibly attacked watermarked image $\acute{f_w}$ were computed using the proposed method. The selection of moments in extraction and embedding processes were similar. Thus, the PCET/QPCET moments which are represented by the vector, $\Phi^*(l) = \left\{\Phi^*_{m_1 n_1}, \Phi^*_{m_2 n_2}, \ldots\ldots\ldots, \Phi^*_{m_l n_l}\right\}$, of the watermarked image $\acute{f_w}$ can be selected by the same key K as in the watermark embedding procedure that carry the watermark information.

2) BINARY WATERMARK EXTRACTION

Using the same Δ and quantizer as in the embedding process given by (14), the magnitude of each Φ'_i is quantized with the two dithers, respectively:

$$\left|\Phi'(l)\right|_j = \left[\frac{|\Phi^*(l)| - d_k(j)}{\Delta}\right] * \Delta + d_k(j), \quad j = 0, 1 \quad (19)$$

By comparing the distances between $\Phi'(l)$ and its quantized version $|\Phi^*(l)|$, we receive:

$$\hat{b}_2(l) = \underset{j\in[0,1]}{\operatorname{argmin}} \left(\left|\Phi'(l)\right|_j - \left|\Phi^*(l)\right|\right)^2 \quad (20)$$

where $\left|\Phi'(l)\right|_j$ is the ith PCET/QPCET moments of the attacked medical image which is quantized considering a bit value of $j \in [0, 1]$. Therefore, a bit $\hat{b}_2(l)$ is decided to be 0 or 1 according to the distance between the corresponding quantized PCET/QPCET moments and its original value $\Phi^*(l)$. The extracted bit is assigned depending on the j of the minimum distance $\hat{b}_2(l)$ value. The one-dimensional binary sequence was extracted from PCET/QPCET moments magnitudes from [29], where $\hat{B}_2 = \left\{\hat{b}_2(l) \in \{0, 1\}, 0 \le l < P \times Q\right\}$ was transformed to represent the binary watermark image. As a result, the binary image was descrambled by using inverse Arnold transform to form the binary watermark image as $\hat{B}_1 = \left\{\hat{b}_1(i, j) \in \{0, 1\}, 0 \le i < P, 0 \le j < Q\right\}$. Finally, the flowchart of the proposed medical image watermarking is depicted in Fig. 1.

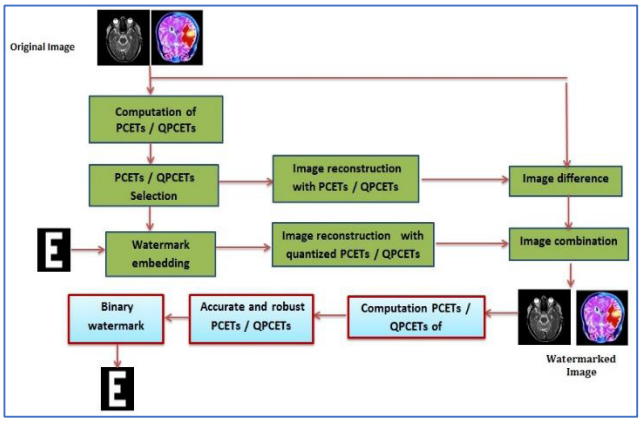

FIGURE 1. The flowchart of the proposed watermarking process.

C. PARALLEL COMPUTATION OF ACCURATE PCET/QPCET MOMENTS

Watermarking of gray-level and color medical images using PCET/QPCET moments require accurate computation of these moments. The highly accurate and numerically stable moments are essentials for the watermarking process to be robust against the common signal processing attacks. Also, highly accurate and numerically stable RST moment invariants ensure the robustness of the watermarking process against the geometric distortions. Since the Polar Harmonic Transforms and their quaternion models are defined in polar coordinates over a unit circle, the entire computational processes will be done in the polar coordinates using polar raster.

Recently, Hosny and Darwish [49] proposed a novel kernel-based highly accurate and numerically stable method for computing the PCET moments. Similar remarkable algorithms were proposed in [50] and [51] for highly accurate and numerically stable QPCET moments for color images. In this subsection, a parallel implementation of these accurate method using CUDA and OpenMP for GPU and multi-core CPUs, respectively, is presented.

1) IMAGE WATERMARKING PROFILING

In order to understand the computational weight of each step of the watermarking process, the sequential implementation is profiled. Table 1 list these steps alongside their execution times for a gray-level and color medical images of sizes 256×256 and PCET/QPCET moments of a maximum order equal to (80, 80).

TABLE 1. Sequential PCET and QPCET watermarking profiling in seconds for moment order (80,80).

Step	Time in seconds	
	Gray-Level Image	Color Image
Image Interpolation	0.560	0.560
Kernels Computation, Ip & Iq,	2.700	2.700
Moment computation	48.100	144.312
embedding watermark	0.001	0.001
Reconstructing the watermarked image	78.550	235.650
Total time	126.912	383.211

Apparently, the most two intensive-computing steps of the PCET/QPCET moment-based watermarking algorithm are: (1) The moment computation, (2) The reconstruction of the watermarked image after embedding the watermark. These two steps are compute-intensive tasks due to the involved huge number of arithmetic operations. The computation of the PCET/QPCET moments using equations (1) and (11) and reconstructing the watermarked image using equations (6) and (13) include scanning each moment value for each image sector (polar pixel); thus, the time complexity of these two operations are $O(N^2 \times MAX^2)$, where N^2 represents the total number of pixels and $MAX = n\_Max = m\_Max$.

2) PARALLEL IMAGE WATERMARKING ON MULTI-CORE CPU

Computing PCET/QPCET moments for medical images of size $n\_Max \times m\_Max$, we distributed the n_Max rows over the available cores. Fig. 2 illustrates this distribution by assigning two rows for each CPU core. In the proposed implementation, the number of PCET/QPCET moment' rows equal n_Max/p, where p is the number of CPU cores. In the same manner, to reconstruct an $N \times N$ image, the total number of pixels is distributed over the available number of CPU cores, p.

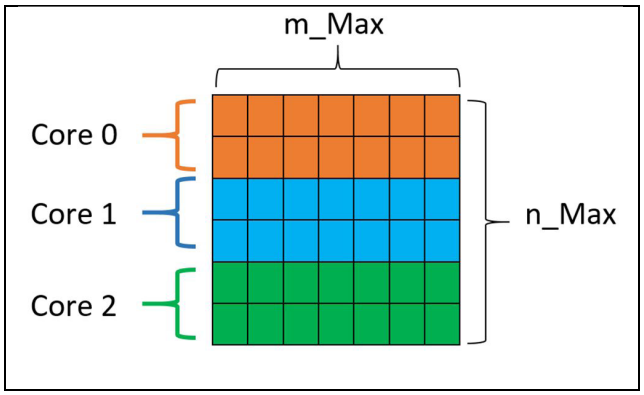

FIGURE 2. Moment distribution over three CPU Cores.

Into the bargain, the other steps of the watermarking are parallelized using the multi-core CPU. For example, the computation of image interpolation, the kernels, I_p and I_q, as defined in [39], steps are parallelized despite their trivial weights as shown in Table 1. Thus, for low moment order these steps are relatively large. In addition, parallelizing these steps contribute to the overall speedup of the watermarking process.

3) PARALLEL IMAGE WATERMARKING ON MANY-CORE GPU

The process of parallelizing image watermarking on a GPU includes two steps, namely, PCET/QPCET moments' computation and watermarked image reconstruction. For moment computation, the elements of the moments are divided over GPU blocks and in turn, these elements are assigned to threads of the block. Each block should access to the interpolated image, the pre-calculated kernel, I_p and I_q, values. As these items are accessible by the entire block, then they are stored in the global memory. In addition, the final computed moment is stored in the global memory as well, as each block should store its contribution to the final computed PCET/QPCET moment. Fig. 3 depicts this process.

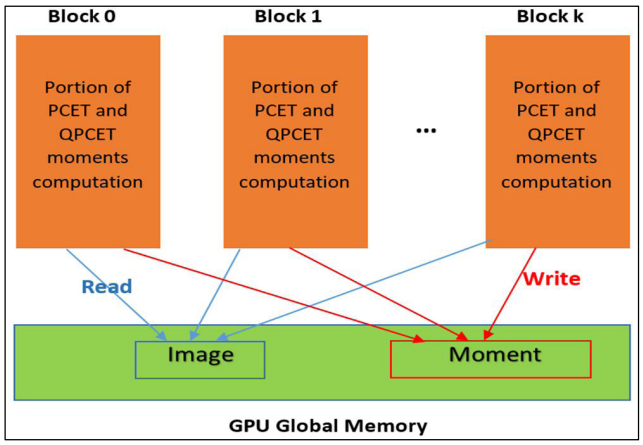

FIGURE 3. Moment computation in GPU.

The process of reconstructing the watermarked image includes computing the pixel value from the computed PCET/QPCET moment, the aforementioned step. The computations of these reconstructed pixels are independent. Thus, parallelizing the watermarked image reconstruction can be achieved by dividing these pixels computations over GPU blocks.

The computations of these pixels require only access to the computed PCET/QPCET moment and store pixels values to the final reconstructed image. Of note, if the image type is gray-scale, then there is one reconstructed value per pixel; otherwise there three values per pixel, for three different channels of the color image. Fig. 4 illustrates this process. In Fig.s 3 and 4 the read and write operations are represents by blue and red arrows, respectively, where the GPU implementation uses k blocks.

4) PARALLEL IMAGE WATERMARKING IMPLEMENTATION DETAILS

In this section, we explain the details of the proposed parallel algorithms to achieve the parallel medical image watermarking task.

From Table 1, the most two compute intensive tasks are the moment computation and the reconstruction of the watermarked image. Thus, we explain the implementation details

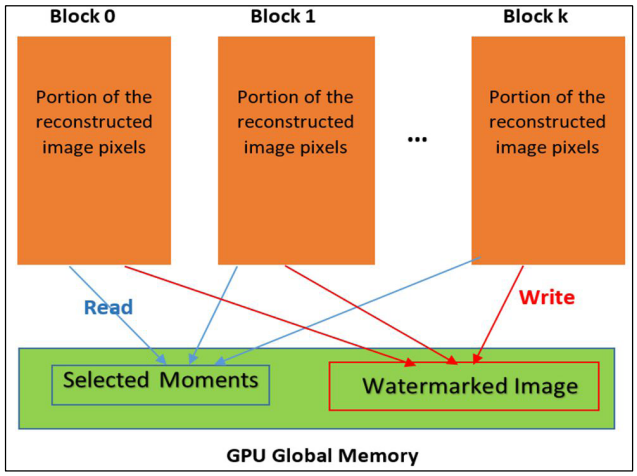

FIGURE 4. Reconstructing watermarked image in GPU.

Algorithm 1: Pseudo-Code of the Parallel Moment Computation

Parallel PCET/PCT
INPUT:
n_Max: number of moment rows
m_Max: number of moment columns
rows: number of images
cols: number of images
PROCDURE:

```
1  Moment_size = n_Max X m_Max
2  in parallel, for i = 1 to image_size
3  |   moment_row = i / m_Max
4  |   moment_col = i MOD m_Max
5  |   for ring_id = 1 to rows
6  |   |   for sector_id = 1 to cols
7  |   |   |   result = computation of Eq. 8
8  |   |   end for
9  |   end for
10 |   M_moment_ow,moment_col = result
11 end for
```

Algorithm 2: Pseudo-Code of the Parallel Watermarked Image Reconstruction

Parallel watermarked image reconstruction
INPUT:
$\mathbf{M_{m\_Max,n\_MAX}}$: the computed moment
n_Max: number of moment rows
m_Max: number of moment columns
N: number of image rows
PROCDURE:

```
1  image_size = N × N
2  in parallel, for i = 1 to image_size
3  |   row = i / N
4  |   col = i MOD N
5  |   for i = 1 to n_Max
6  |   |   for j = 1 to m_Max
7  |   |   |   result = computation of Eq. 11 using M_i,j
8  |   |   end for
9  |   end for
10 |   Reconstructed_Image_row,col = result
11 end for
```

of these two tasks through the proposed pseudo codes listed in ALGORITHMs 1 and 2.

From Fig. 2, it is clear that the moment to be computed consists of a set of elements, $n\_Max \times m\_Max$. To compute each element of this moment, each element of the entire pixels of the image contributes with a value according to Eq. 8. As the images contains rows and columns, then each moment element is computed with two nested loops. The idea of ALGORITHM 1 is to join the first two loops of the moment so that the code has abundant of independent tasks; this yields exactly $n\_Max \times m\_Max$ independent tasks. For example, for $n\_Max = m\_Max = 80$, the code has 6,400 independent tasks. Then, these tasks are distributed on the CPU or GPU cores. These algorithms are for one channel only. Thus, it executed one time for the gray-level images and three times for the color images, as they have three channels.

In ALGORITHM 1, line 1 computes the moment size, as a multiplication of *n_Max* and *m_Max*. Then, lines 2 to 11 present the parallel loop. Each iteration of this loop can be executed in independently. As line 2 joined all the elements of the moment, lines 3 and 4 should determine which element of the moment is computed per each iteration. Thus, lines 3 and 4 computes the moment element row and column values; these values are used to store the computed moment element in the moment array, line 10.

Lines 5 to 9 of ALGORITHM 1 represent Eq. 11. Thus, these lines have two loops to iterate through each pixel of the image pixels. Similarly, the algorithm of reconstructing the watermarked medical image has four nested loops. Each iteration of the first two loops represent a single image pixel; the inner two loops represent the computation required to reconstruct the iteration pixel.

ALGORITHM 2 lists the steps of the reconstructing of the medical watermarked image. The algorithm has three loops, as the first two loops of image rows and columns are joined to increase the number of independent tasks, lines 2 to 11. Lines 3 and 4 compute the row and column value of the reconstructed image pixel. Then, lines 5 to 9 do the computation of Eq. 13 using the moment element $M_{i,j}$. Finally, the reconstructed pixel is stored in the final reconstructed image.

V. NUMERICAL EXPERIMENTS

A. TEST IMAGES AND EVALUATION MEASURES

The performance of the watermarking approach for gray-level and color medical images is evaluated by the watermark invisibility, watermark imperceptibility, robustness of

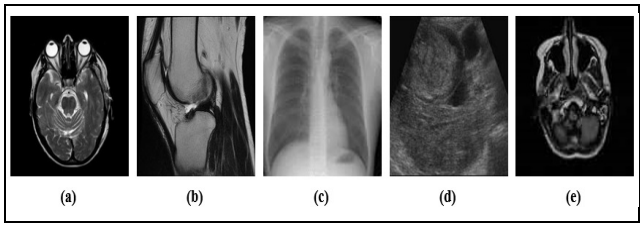

FIGURE 5. Samples of the gray medical images used in Testing the watermarking methods.

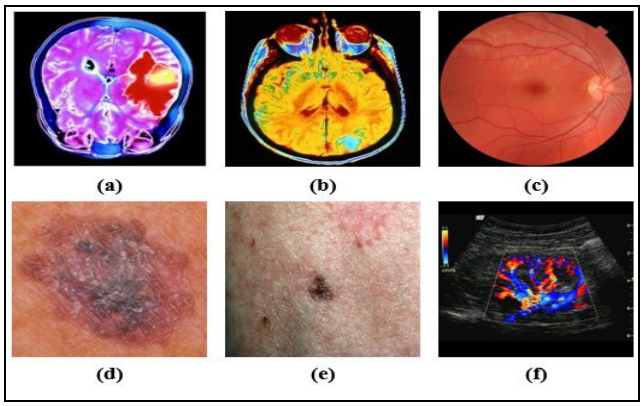

FIGURE 6. Samples of Color medical images used in Testing the watermarking methods.

the watermark, and computational complexity. The watermark invisibility is measured by using the peak signal-to-noise ratio (PSNR) and the structural similarity image index (SSIM), while the robustness of watermark approach is evaluated by using the bit error rate (BER). Moreover, the normalized correlation (NC) is used to provide an objective judgment of the robustness of extracted watermark.

A set of 256 × 256 gray-level and color medical images as displayed in Figs. 5 and 6, are used as the host images. Binary image of size 32 × 32 as shown in Figs. 7.a–7.I, are used as original watermarks.

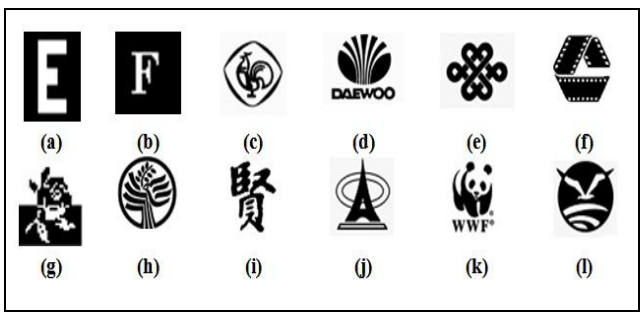

FIGURE 7. Samples of binary images used in Testing the watermarking methods.

For fair comparison, the proposed PCET/QPCET methods are compared with the DCT/QDCT watermarking algorithms where the codes are implemented with the entire images to avoid the blocking artifacts and executed using the same machine.

The implementations of the proposed watermarking algorithm are written in C++. OpenMP threads library and CUDA 7.5 are utilized for the multi-core CPU and GPU programming, respectively. The utilized OS is 64-bit Linux. The experiments are conducted on a computer with two 2.3 GHz Intel 8-core processors with 32 GB RAM and K20m NVIDIA GPU.

1) THE PSNR AND SSIM MEASURE

For measuring the invisibility of the watermarked images, two metrics that includes (PSNR) and (SSIM) are adopted to evaluate the quality of the watermarked image by measuring the objective similarity between the original host image f and the watermarked image f_w. A larger value of PSNR indicates that the watermarked image is more closely resembles to the original medical image which meaning that the watermark is more imperceptible. The PSNR is defined and calculated as follows:

$$PSNR\left(f,f_w\right)=10\log_{10}\frac{255^2}{MSE} \tag{21}$$

The mean square error (MSE) is defined as follows:

$$MSE=\frac{1}{N^2}\left(\sum_{i=1}^{N}\sum_{j=1}^{N}\left[f_w\left(i,j\right)-f\left(i,j\right)\right]^2\right) \tag{22}$$

Additionally, we also include the structural similarity (SSIM) [52] index to measure the similarity between the original image f and the embedded image f_w. The structural similarity (SSIM) index is considered to be correlated with the quality perception of the human visual system (HVS). It is used to evaluate the similarity between the original image f and the watermarked image f_w.

The values of SSIM are falling in the interval, [0, 1]. When SSIM value is 0, it means that $f \neq f_w$. When SSIM value is 1, it means that $f = f_w$. The measure SSIM is defined as follows:

$$\mathrm{SSIM}\left(f,f_w\right)=l\left(f,f_w\right)c\left(f,f_w\right)s\left(f,f_w\right) \tag{23}$$

where $\boldsymbol{l}\left(\boldsymbol{f},\boldsymbol{f_w}\right)$ is the luminance comparison function, $\boldsymbol{c}\left(\boldsymbol{f},\boldsymbol{f_w}\right)$ is the contrast comparison function, and $\boldsymbol{s}\left(\boldsymbol{f},\boldsymbol{f_w}\right)$ is the structure comparison function.

2) THE BER MEASURE

The robustness of the watermark is measured quantitatively by the BER of the extracted watermark. The BER [53] is defined as the ratio between the number of incorrectly extracted bits and the total number of bits, its mathematical definition is:

$$\mathrm{BER}=\frac{\mathrm{B_{error}}}{\mathrm{l}} \tag{24}$$

where $\boldsymbol{B_{error}}$ is the number of erroneously extracted bits, while $\boldsymbol{l}=\boldsymbol{P}\times\boldsymbol{Q}$ is the total number of embedded bits or the watermark image dimensions. BER is a quantitative measure used to test and evaluate the robustness of the watermarking

approaches against geometric and common signal processing attacks. The value 0 is the ideal value of BER.

3) THE NC MEASURE

The robustness of the proposed watermarking algorithm is determined in term of a correlation factor. The similarity and differences between the original and the extracted watermark are measured by the Normalized Correlation (NC) [54]. The NC is a quantitative measure used to determine the robustness of watermark against different attacks. Moreover, its value is generally ranging from 0 to 1, NC $\in$ [0, 1]. The normalized correlation (NC) between the original watermark W and the extracted watermark w$''$ is calculated as follows:

$$NC = \frac{\sum_{i=1}^{N}\sum_{j=1}^{N}\left[W\,(i,j)\,xw''(i,j)\right]}{\sum_{i=1}^{N}\sum_{j=1}^{N}\left[W\,(i,j)\right]^2} \tag{25}$$

when the value of NC is close to 1, it means that the watermarking method is robust against the attacks.

B. THE WATERMARK INVISIBILITY ANALYSIS

Experiments were performed to evaluate the watermark invisibility of different host medical images. In the conducted experiments, each of the test watermarks are embedded in the test gray-level and color medical images using the proposed PCET/QPCET and the existing DCT/QDCT watermarking algorithms. The binary watermarks are resized to the size 10×10 where the total number of embedded bits is 100. In the proposed methods, each binary watermark is embedded in the host gray-level and color medical images respectively with different quantization step $\mathbf{\Delta}$. As a result, the gray-level and the color watermarked medical images are tested with all binary watermark images. For each quantization step $\mathbf{\Delta}$, the values of the PSNR and SSIM are computed where the average values of PSNR and SSIM are listed in Table 2.

TABLE 2. The average PSNR (dB) & SSIM values for watermarking approaches with different values of quantization step (Δ).

	Proposed Methods				Discrete Cosine Transform			
Quant.	PCET		QPCET		DCT		QDCT	
step	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM
0.2	53.64	0.980	53.91	0.999	52.13	0.980	52.04	0.996
0.4	47.592	0.966	47.75	0.996	46.62	0.963	46.90	0.986
0.6	43.589	0.950	44.76	0.994	42.07	0.945	43.46	0.973
0.8	41.55	0.950	42.33	0.991	40.52	0.905	41.35	0.959
1.0	40.597	0.933	41.35	0.989	39.65	0.862	39.61	0.946

Additionally, the average PSNR values for a color watermarked medical image are computed by the proposed QPCET and the existing QDCT are displayed in Fig. 8. It is clear that the performance of the proposed QPCET watermarking is better than the performance of the existing QDCT watermarking algorithm where the average PSNR value is increasing with decreasing the quantization step, which ensure the strength of the watermark invisibility of the proposed algorithm.

Table 2 clearly shows that the proposed PCET/QPCET watermarking algorithm outperforms the corresponding

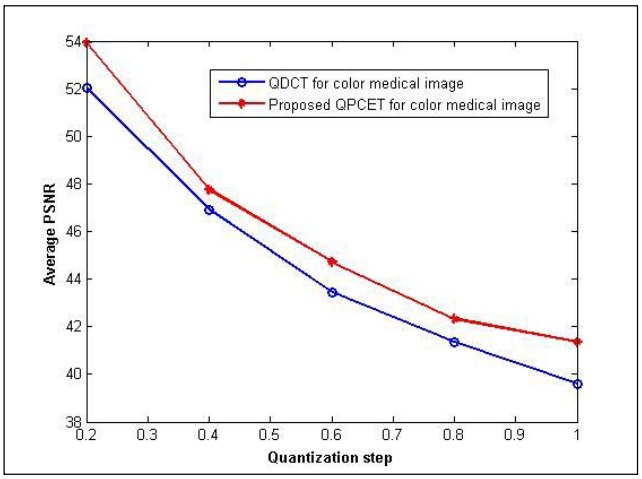

FIGURE 8. Visual imperceptibility relative to quantization step Δ.

DCT/QDCT watermarking algorithms and achieves excellent results in terms of PSNR and SSIM.

Based on Fig. 8 and Table 2, the quantization step plays an important role in evaluating watermark invisibility. Therefore, the appropriate choice of the embedding quantization step $\mathbf{\Delta}$ produce a better visual imperceptibility. Experiments show that the quantization step (Δ) equal to 0.2 is an excellent choice. Thus, this step value will be used through the entire conducted experiments.

A qualitative evaluation of the proposed PCET/QPCET watermarking algorithms is performed where the watermarked gray-level and color medical images are reconstructed where the reconstructed images are displayed in Fig. 9 and Fig. 10, respectively.

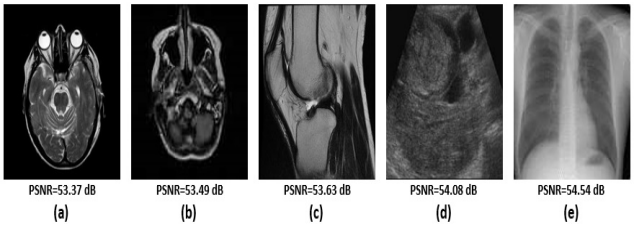

FIGURE 9. The watermarked image for the gray medical image of size 256X256 pixel with $\Delta = 0.2$.

The obtained high quality reconstructed watermarked medical images are due to the highly accurate computational method which utilized exact kernels to compute PCET/QPCET moments without any approximation errors. Stability of the computed PCET/QPCET moments enable us to use higher order moments for perfect reconstruction of the watermarked medical images. No distortion is observed in the reconstructed watermarked medical images.

The watermarked color images with large PSNR values have great imperceptibility and very close to the original ones in visual quality. This means that the proposed

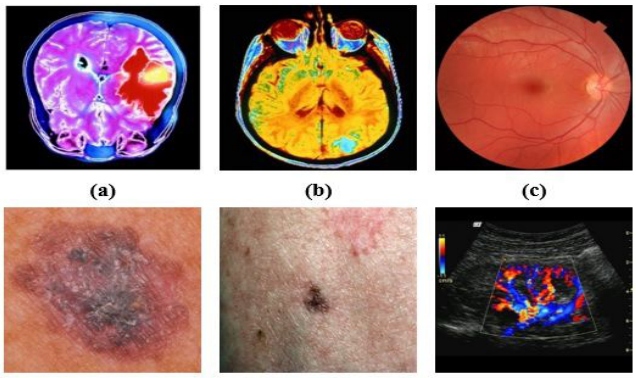

FIGURE 10. The watermarked image for the color medical images of size 256X256 pixel with $\Delta = 0.2$.

PCET/QPCET-based watermarking algorithms achieve great imperceptibility.

C. THE WATERMARK ROBUSTNESS ANALYSIS

In order to investigate and evaluate the robustness of the proposed watermarking algorithm, various geometric and signal processing attacks have been applied on the watermarked medical images. These attacks include the JPEG compression, contamination with different kinds of additive noise, image filtering, image rotation, image scaling, and image translation. A set of numerical experiments are performed where the watermark binary images are embedded in host medical images and extracted using the proposed PCET/QPCET and the existing DCT/QDCT algorithms. These medical images are attacked by different individual and multiple attacks.

In Table 3, the watermark binary image "F" is extracted by using the proposed PCET/QPCET and DCT/QDCT for gray-level and color medical images. For each attack, the extracted binary images are displayed. It is clear that no distortion can be seen in the images extracted by the proposed method which ensure the robustness of the proposed PCET/QPCET watermarking algorithm against the common attacks. On the other side, many distortions are noticed in the images extracted by the DCT/QDCT watermarking methods.

The BER and NC values are quantitative measures used to evaluate the extracted watermarks. The obtained BER and NC values are presented in Tables 4 and 5, respectively.

Since the ideal value of BER is 0; Table 4 clearly show that for all attacks, the obtained values of BER are zeros or approaching zero which ensure the robustness of the proposed PCET/QPCET watermarking algorithms. On the other side, the BER values for the DCT/QDCT watermarking algorithms are much higher than zero which mean the sensitivity of the DCT/QDCT for different attacks.

The ideal value of NC is 1. The obtained values of NC are shown in Table 5. It is observed that the NC values of the PCET/QPCET watermarking algorithms are approaching 1 and the minimum value is 0.9. These values ensure the robustness of the proposed PCET/QPCET watermarking algorithm against the geometric and common signal processing attacks. The DCT/QDCT watermarking methods are very

TABLE 3. The extracted binary watermark under common attacks.

Attacks		Proposed methods		Discrete Cosine Transform	
		PCET for gray images	QPCET for color images	DCT for gray images	QDCT for color images
	10°	F	F		
	15°	F	F		
	20°	F	F		
	25°	F	F		
	30°	F	F		
	35°	F	F		
	40°	F	F		
	45°	F	F		
Scaling factor	0.75	F	F		
	1.25	F	F		
	1.5	F	F	F	F
	1.75	F	F	F	F
	2	F	F	F	F
Translation	(H 2, V 2)	F	F	F	F
	(H 5, V 5)	F	F	F	F
Scaling 1.5 + Rotation 15°		F	F	F	F
Scaling 1.5 + Rotation 25°		F	F		
Scaling 0.75 + JPEG compression (90%)		F	F		F
Scaling 1.5 + JPEG compression (90%)		F	F	F	F
Scaling 2.0 + JPEG compression (90%)		F	F	F	F
Rotation 35° + JPEG compression (90%)		F	F		
Rotation 45° + JPEG compression (90%)		F	F		
JPEG compression	50	F	F	F	F
	70	F	F	F	F
	90	F	F	F	F
Salt & Peppers Noise	0.01	F	F	F	F
	0.05	F	F	F	F
Gaussian noise	0.01	F	F	F	F
	0.05	F	F	F	F
Gaussian Filtering	(3*3)	F	F	F	F
	(5*5)	F	F	F	F
Median Filtering	(3*3)	F	F	F	F
	(5*5)	F	F	F	F

TABLE 4. The BER values for the extracted watermark under signal processing and geometric attacks.

Attacks		Proposed methods		Discrete Cosine Transform	
		QPCET For color images	PCET For Gray images	QDCT For color images	DCT For Gray images
Rotation angles	5°	0.0059	0.0029	0.1709	0.2158
	10°	0.0039	0.0010	0.1299	0.2207
	15°	0.0010	0.0010	0.1514	0.2295
	20°	0.0049	0.0020	0.1191	0.2334
	25°	0.0049	0.0039	0.1475	0.2422
	30°	0.0049	0.0029	0.1230	0.2314
	35°	0.0068	0.0020	0.1563	0.2334
	40°	0	0.0029	0.1299	0.2412
	45°	0.0039	0.0020	0.1650	0.2412
Scaling factor	0.75	0.0010	0.0020	0.0410	0.1035
	1.25	0	0.0020	0.0264	0.0117
	1.5	0.0010	0.0010	0.0117	0.0146
	1.75	0.0020	0.0029	0.0146	0.0137
	2	0	0.0010	0.0186	0.0176
Translation	(H 2, V 2)	0	0.0010	0.0176	0.0166
	(H 5, V 5)	0.0010	0.0010	0.0332	0.0303
Scaling 1.5 + Rotation 15°		0.0010	0	0.1504	0.2617
Scaling 1.5 + Rotation 25°		0.0020	0	0.1543	0.2607
Scaling 0.75 + JPEG compression (90%)		0.0156	0.0068	0.0176	0.1611
Scaling 1.5 + JPEG compression (90%)		0	0.0010	0.0146	0.0225
Scaling 2.0 + JPEG compression (90%)		0.0010	0.0020	0.0186	0.0342
Rotation 35° + JPEG compression (90%)		0.0049	0.0020	0.2217	0.2529
Rotation 45° + JPEG compression (90%)		0.0107	0.0059	0.2422	0.2705
JPEG Compression	50	0.0137	0.0176	0.0947	0.0693
	70	0.0029	0.0010	0.0723	0.0264
	90	0.0010	0	0.0068	0.0020
Salt and Peppers Noise	0.01	0.0020	0.0010	0.0225	0.0205
	0.05	0.0049	0.0029	0.0430	0.0371
Gaussian noise	0.01	0.0010	0	0.0234	0.0176
	0.05	0.0029	0.0020	0.0313	0.0254
Gaussian Filtering	(3*3)	0.0049	0.0029	0.0430	0.0332
	(5*5)	0.0098	0.0059	0.0693	0.0596
Median Filtering	(3*3)	0.0010	0.0010	0.0313	0.0293
	(5*5)	0.0029	0.0010	0.0693	0.0537

TABLE 5. NC values of the extracted watermarks.

Attacks		Proposed methods		Discrete Cosine Transform	
		QPCET For color image	PCET For gray image	QDCT For color images	DCT For gray images
Rotation	5°	0.9661	0.9823	0.4822	0.4084
	10°	0.9762	0.9941	0.5455	0.4548
	15°	0.9941	0.9940	0.5157	0.4079
	20°	0.9704	0.9880	0.6016	0.4050
	25°	0.9715	0.9762	0.5258	0.3782
	30°	0.9715	0.9820	0.6048	0.4014
	35°	0.9577	0.9881	0.5346	0.3948
	40°	1.0	0.9821	0.5605	0.3840
	45°	0.9770	0.9883	0.5142	0.4242
Scaling factor	0.75	0.9940	0.9880	0.7971	0.6333
	1.25	1.0	0.9880	0.8238	0.9341
	1.5	0.9940	0.9940	0.9258	0.9179
	1.75	0.9880	0.9820	0.9063	0.9243
	2.0	1.0	0.9940	0.8797	0.9038
Translation	(H 2, V 2)	1.0	0.9941	0.8864	0.8931
	(H 5, V 5)	0.9941	0.9940	0.8437	0.8465
Scaling 1.5 + Rotation 15°		0.9940	1	0.5118	0.3654
Scaling 1.5 + Rotation 25°		0.9881	1	0.5017	0.3910
Scaling 0.75 + JPEG Compression (90%)		0.9100	0.9599	0.8864	0.5087
Scaling 1.5 + JPEG Compression (90%)		1.0	0.9940	0.9063	0.8794
Scaling 2.0 + JPEG Compression (90%)		0.9940	0.9880	0.8797	0.8402
Rotation 35° + JPEG Compression (90%)		0.9883	0.9709	0.4191	0.4012
Rotation 45° + JPEG Compression (90%)		0.9643	0.9358	0.4186	0.3600
JPEG Compression	50	0.9189	0.9020	0.6410	0.7075
	70	0.9826	0.9940	0.7168	0.8699
	90	0.9940	1	0.9586	0.9883
Salt and Peppers	(0.01)	0.9880	0.9940	0.8816	0.8903
	(0.05)	0.9698	0.9823	0.8035	0.8206
Gaussian noise	(0.01)	0.9940	1	0.8773	0.9003
	(0.05)	0.9820	0.9698	0.8482	0.8666
Gaussian Filtering	(3*3)	0.9698	0.9820	0.7902	0.8437
	(5*5)	0.9431	0.9636	0.6864	0.7261
Median Filtering	(3*3)	0.9940	0.9941	0.8399	0.8531
	(5*5)	0.9821	0.9940	0.6991	0.7700

sensitive to the geometric attacks where the NC values are dramatically fall to 0.40 when the images rotated with acute angles. The values of NC for the DCT/QDCT watermarking algorithms are generally low which ensure the sensitivity of the DCT/QDCT algorithms to all kinds of attacks.

Additional numerical experiments are performed where the binary watermark images as displayed in Fig. 7 are resized to 10×10 and embedded in the host gray-level and color medical images by using the proposed PCET/QPCET and the DCT/QDCT watermarking algorithms, where the watermarked images are generated. For each watermarked image, each watermark is extracted under geometric distortions and different common attacks. The average values of BERs of the extracted watermarks are listed in Table 6.

Based on the results in Table 6, we can see that the proposed PCET/ QPCET watermarking algorithms have much robustness to all geometric and image processing attacks in comparison to DCT/QDCT algorithms. Moreover, most of BER values are zero and this mean that extraction of watermarks from watermarked medical images are highly accurate and more robustness against various attacks. It is clear that the proposed watermarking algorithm successfully resisting the sever individual and concatenated geometric attacks and common signal processing attacks with different levels of severity.

Generally, Tables 3 to 6 show that the proposed PCET/ QPCET watermarking algorithms outperform the DCT/ QDCT watermarking algorithms in terms of accuracy.

D. THE PARALLEL WATERMARKING RUNNING TIME ANALYSIS

Fast watermarking of medical images is an essential counterpart in securing the medical images. To accelerate

TABLE 6. Average BER values of the watermarking methods for various attacks.

Attacks		Proposed methods		Discrete Cosine Transform	
		PCET for gray images	QPCET for color images	DCT For gray images	QDCT For color images
Rotation angle	10°	0	0	0.0176	0.0143
	20°	0	0	0.0203	0.0182
	30°	0	0	0.0220	0.0203
	40°	0	0	0.0158	0.0127
	45°	0	0	0.0113	0.0098
Scaling factor	0.25	0.0350	0.0200	0.2204	0.2016
	0. 5	0	0	0.0604	0.0485
	0.75	0	0	0.0205	0.0189
	1.25	0	0	0.0113	0.0106
	1.5	0	0	0.0100	0.0084
	1.75	0	0	0.0074	0.0058
	2.0	0	0	0.0029	0.0020
Translation	(H 2, V 2)	0	0	0.0074	0.0059
	(H 4, V 4)	0	0	0.0098	0.0084
	(H 5, V 2)	0	0	0.0104	0.0098
	(H 5, V 10)	0.0029	0.0020	0.0138	0.0117
	(H 5, V 15)	0.0059	0.0039	0.0193	0.0175
JPEG Compression	30	0.0400	0.0200	0.4303	0.4226
	40	0.0098	0.0059	0.1200	0.1100
	50	0	0	0.0800	0.0600
	70	0	0	0.0200	0.0100
	90	0	0	0.0098	0.0049
Shearing (1%-2%)		0.0098	0.0074	0.0127	0.0115
Scaling 1.25 + Rotation 25°		0	0	0.0105	0.0098
Scaling 1.5 + Rotation 35°		0	0	0.0303	0.0205
Scaling 0.75 + JPEG Compression (90%)		0	0	0.0196	0.0167
Scaling 1.5 + JPEG Compression (90%)		0	0	0.0127	0.0105
Rotation 15° + JPEG Compression (90%)		0	0	0.0224	0.0200
Rotation 35° + JPEG Compression (90%)		0	0	0.0313	0.0276
Gaussian Noise (0.02) + JPEG compression (90%)		0.0029	0.0010	0.0137	0.0117
Transl. (H 5, V 10) + Salt & Peppers Noise (0.02)		0.0039	0.0020	0.0130	0.0121
Salt & Peppers Noise (0.02)		0.0049	0.0039	0.0142	0.0129
Gaussian Noise (0.02)		0.0029	0.0020	0.0132	0.0124
Median Filtering (3×3)		0.0010	0.0029	0.0117	0.0102

TABLE 7. Speedups of Parallel Watermarking against sequential version for PCET/QPECT and a 256 × 256 image and the moment order 60.

	Color Medical Images			Gray-Level Medical Image		
	Time	Speedup	Improv.	Time	Speedup	Improv.
Sequential	180.1	--	--	60.5	---	----
2 cores	94.3	1.91x	47.62%	31.6	1.92x	47.78%
4 cores	49.2	3.67x	72.68%	16.5	3.67x	72.73%
8 cores	28.3	6.36x	84.29%	9.6	6.30x	84.13%
16 cores	16.3	11.05x	90.95%	5.5	11.00x	90.91%
GPU	8.86	20.33x	95.08%	3.2	18.91x	93.71%

TABLE 8. Speedups of parallel watermarking against sequential version for DCT/QDCT and a 256 × 256 image and the moment order 60.

	Color Medical Images			Gray-level Medical Image		
	Time	Speedup	Improv.	Time	Speedup	Improv.
Sequential	607.3	--	--	204.0	--	--
2 cores	312.7	1.90x	48.52%	104.8	1.90x	48.64%
4 cores	156.6	3.90x	74.21%	52.5	3.90x	74.26%
8 cores	85.6	7.10x	85.90%	29.1	7.00x	85.76%
16 cores	44.8	13.50x	92.62%	15.1	13.50x	92.58%
GPU	55.0	11.00x	90.95%	19.9	10.30x	90.27%

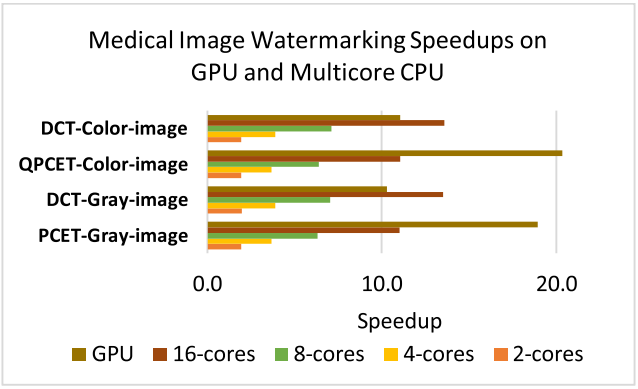

FIGURE 11. Parallel Watermarking Speedups on Multicore CPU and GPU.

the sequential watermarking algorithm for gray-level and color medical images, a parallel implementation by using multi-core CPUs and K20m GPU architectures are performed, Tables 7 and 8.

Table 7 shows the elapsed times for watermarking of different gray-level and color medical images of size 256×256 and moment order equals $(60, 60)$. Theoretically, the optimal speedup of 2, 4, 8, and 16 CPU cores are $2\times$, $4\times$, $8\times$, and $16\times$, respectively. Practically, these values could not be reached. The obtained speedups for PCET/QPCET are $1.91\times$, $3.76\times$, $6.37\times$, and $11.05\times$, for 2, 4, 8, and 16 CPU cores, respectively.

In addition to the speedup, two others measures are used. The first measure is the explicit running times in seconds. The second, is the execution time improvement ratio [55] which is a percentage used to compare two execution times. The ETIR is calculated by using the following form:

$$\left(\frac{sequential_{time} - parallel_{time}}{sequential_{time}}\right) \times 100\%$$

The obtained values of these measures for the PCET/QPCET watermarking algorithms are shown in Table 7. Similar parallel implementations for DCT/QDCT using the same machine are performed. The obtained speedups, the explicit running times and ETIR values are shown in Table 7 and 8.

The parallel speedups of the PCET/QPCET and DCT/QDCT watermarking algorithms on Multi-Core CPUs and GPU are displayed in Fig. 11. Finally, the explicit running times in seconds for the proposed PCET/QPCET and DCT/QDCT, for a gray-level image of size 256×256, using

TABLE 9. Execution times of parallel watermarking of the propose method against DCT method for different moment orders using a GPU.

	Proposed method		Discrete Cosine Transforms	
Moment order	PCET	QPCET	DCT	QDCT
20	2.2	6.1	10.8	29.9
40	2.9	8.0	14.1	38.9
60	3.2	8.9	19.9	55.0

the K20m GPU are shown in Table 9. This table shows a considerable time difference between the two approaches on a powerful GPU unit.

Based on the speedups, the explicit running times and the ETIR, it is observed that the PCET/QPCET watermarking algorithms are much faster than their corresponding DCT/QDCT watermarking algorithms. Generally, the entire results show that the proposed PCET/QPCET watermarking algorithms outperform the widely-used DCT/QDCT watermarking algorithms.

VI. CONCLUSION

Securing gray-level and color medical images is achieved using accurate and robust watermarking algorithm. Parallel implementations on multi-core CPUs and GPUs tremendously reduce the watermarking overall time. Watermarking of a single-color medical image using a sequential implementation requires approximately 180 second. This time is reduced to 8 seconds using the proposed implementation on GPU architecture. This significant reduction in time is desirable for real-time applications in medical imaging systems where the medical images could be secured once they are reconstructed from the sensed medical data. Fast securing of the medical images minimize the probability of alter their contents. The door is still open for more future fast algorithms and more reduction in the overall watermarking execution times using different parallel architectures.

Implementation of this fast, accurate and robust watermarking algorithm with circuits to be embedded directly in medical imaging devices is the scope of our future work.

REFERENCES

[1] R. Wootton, J. Craig, and V. Patterson, *Introduction to Telemedicine*, vol. 206. London, U.K.: Royal Society of Medicine Press, 2006.
[2] T. H. F. Broens *et al.*, "Determinants of successful telemedicine implementations: A literature study," *J. Telemed. Telecare*, vol. 13, no. 6, pp. 303–309, 2007.
[3] A. K. Singh, B. Kumar, G. Singh, and A. Mohan, Eds., *Medical Image Watermarking: Techniques and Applications*. Springer, 2017.
[4] A. K. Singh, B. Kumar, M. Dave, and A. Mohan, "Robust and imperceptible dual watermarking for telemedicine applications," *Wireless Pers. Commun.*, vol. 80, no. 4, pp. 1415–1433, 2014.
[5] J. W. K. Gnanaraj, K. Ezra, and E. B. Rajsingh, "Smart card based time efficient authentication scheme for global grid computing," *J. Hum.-Centric Comput. Inf. Sci.*, vol. 3, no. 16, pp. 1–17, Sep. 2013.
[6] M. Iftikhar, H. M. P. Singh, and M. S. Arifianto, "Telecardiology for e-diagnosis and e-learning in rural area of Sabah, Malaysia: A novel approach for cardiac services in rural medicine," in *Proc. Int. Conf. Adv. Hum.-Oriented Personalized Mech., Technol. Services*, Aug. 2010, pp. 83–88.
[7] G. Coatrieux, H. Huang, H. Shu, L. Luo, and C. Roux, "A watermarking-based medical image integrity control system and an image moment signature for tampering characterization," *IEEE J. Biomed. Health Inform.*, vol. 17, no. 6, pp. 1057–1067, Nov. 2013.
[8] S. M. Mousavi, A. Naghsh, and S. A. R. Abu-Bakar, "Watermarking techniques used in medical images: A survey," *J. Digit. Imag.*, vol. 27, no. 6, pp. 714–729, 2014.
[9] A. F. Qasim, F. Meziane, and R. Aspin, "Digital watermarking: Applicability for developing trust in medical imaging workflows state of the art review," *Comput. Sci. Rev.*, vol. 27, pp. 45–60, Feb. 2018.
[10] D. S. Chauhan, A. K. Singh, B. Kumar, and J. P. Saini, "Quantization based multiple medical information watermarking for secure e-health," *Multimedia Tools Appl.*, pp. 1–13, Jun. 2017.
[11] A. K. Singh, "Improved hybrid technique for robust and imperceptible multiple watermarking using medical images," *Multimedia Tools Appl.*, vol. 76, no. 6, pp. 8881–8900.
[12] R. Thanki, S. Borra, V. Dwivedi, and K. Borisagar, "A RONI based visible watermarking approach for medical image authentication," *J. Med. Syst.*, vol. 41, no. 9, p. 143, 2017.
[13] F. N. Thakkar and V. K. Srivastava, "A blind medical image watermarking: DWT-SVD based robust and secure approach for telemedicine applications," *Multimedia Tools Appl.*, vol. 76, no. 3, pp. 3669–3697, 2017.
[14] S. A. Parah, J. A. Sheikh, F. Ahad, N. A. Loan, and G. M. Bhat, "Information hiding in medical images: A robust medical image watermarking system for E-healthcare," *Multimedia Tools Appl.*, vol. 76, pp. 10599–10633, Apr. 2017.
[15] S. Priyanka and S. Maheshkar, "Region-based hybrid medical image watermarking for secure telemedicine applications," *Multimedia Tools Appl.*, vol. 76, pp. 3617–3647, Feb. 2017.
[16] C.-H. Teh and R. T. Chin, "On image analysis by the methods of moments," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. PAMI-10, no. 4, pp. 496–513, Jul. 1988.
[17] Y. Xin, S. Liao, and M. Pawlak, "Circularly orthogonal moments for geometrically robust image watermarking," *Pattern Recognit.*, vol. 40, pp. 3740–3752, Dec. 2007.
[18] I. A. Ismail, M. A. Shouman, K. M. Hosny, and H. M. Abdel-Sala, "Invariant image watermarking using accurate Zernike moments," *J. Comput. Sci.*, vol. 6, no. 1, pp. 52–59, 2010.
[19] H. Q. Zhu, M. Liu, and Y. Li, "The RST invariant digital image watermarking using Radon transforms and complex moments," *Digit. Signal Process.*, vol. 20, no. 6, pp. 1612–1628, 2010.
[20] X.-Y. Wang and L.-M. Hou, "A new robust digital image watermarking based on Pseudo-Zernike moments," *Multidimensional Syst. Signal Process.*, vol. 21, no. 2, pp. 179–196, 2011.
[21] H. Zhang *et al.*, "Affine Legendre moment invariants for image watermarking robust to geometric distortions," *IEEE Trans. Image Process*, vol. 20, no. 8, pp. 2189–2199, Aug. 2011.
[22] C. Singh and S. K. Ranade, "Geometrically invariant and high capacity image watermarking scheme using accurate radial transform," *Opt. Laser Technol.*, vol. 54, pp. 176–184, Dec. 2013.
[23] E. D. Tsougenis, G. A. Papakostas, D. E. Koulouriotis, and V. D. Tourassis, "Performance evaluation of moment-based watermarking methods: A review," *J. Syst. Softw.*, vol. 85, no. 8, pp. 1864–1884, 2013.
[24] W. Chun-Peng, W. Xing-Yuan, and X. Zhi-Qiu, "Geometrically invariant image watermarking based on fast radial harmonic Fourier moments," *Signal Process., Image Commun.*, vol. 45, pp. 10–23, Jul. 2016.
[25] M. Khalid Hosny and M. Mohamed Darwis, "Invariant image watermarking using accurate polar harmonic transforms," *Comput. Elect. Eng.*, vol. 62, pp. 429–447, Aug. 2017.
[26] E. D. Tsougenis, G. A. Papakostas, D. E. Koulouriotis, and E. G. Karakasis, "Adaptive color image watermarking by the use of quaternion image moments," *Expert Syst. Appl.*, vol. 41, no. 14, pp. 6408–6418, 2014.
[27] X.-Y. Wang, P.-P. Niu, H.-Y. Yang, C.-P. Wang, and A.-L. Wang, "A new robust color image watermarking using local quaternion exponent moments," *Inf. Sci.*, vol. 277, pp. 731–754, Sep. 2014.
[28] P.-P. Niu, P. Wang, Y.-N. Liu, H.-Y. Yang, and X.-Y. Wang, "Invariant color image watermarking approach using quaternion radial harmonic Fourier moments," *Multimedia Tools Appl.*, vol. 75, no. 13, pp. 7655–7679, 2015.
[29] H.-Y. Yang, X.-Y. Wang, P.-P. Niu, and A.-L. Wang, "Robust color image watermarking using geometric invariant quaternion polar harmonic transform," *ACM Trans. Multimedia Comput. Commun. Appl.*, vol. 11, no. 3, pp. 1–26, 2015.

[30] M. Khalid Hosny and M. Mohamed Darwish, "Robust color image watermarking using invariant quaternion Legendre-Fourier moments," *Multimedia Tools Appl.*, vol. 77, no. 19, pp. 24727–24750, Jan. 2018, doi: 10.1007/s11042-018-5670-9.
[31] B. Fang, G. Shen, S. Li, and H. Chen, "Techniques for efficient DCT/IDCT implementation on generic GPU," in *Proc. IEEE Int. Symp. Circuits Syst.*, Kobe, Japan, May 2005, pp. 1126–1129.
[32] M. S. Islam, C.-H. Kim, and J.-M. Kim, "A GPU-based (8, 4) Hamming decoder for secure transmission of watermarked medical images," *Cluster Comput.*, vol. 18, no. 1, pp. 333–341, 2015.
[33] H. L. Khor, S.-C. Liew, and J. M. Zain, "Parallel digital watermarking process on ultrasound medical images in multicores environment," *Int. J. Biomed. Imag.*, vol. 2016, Jan. 2016, Art. no. 9583727, doi: 10.1155/2016/9583727.
[34] A. A. Mohammad and A. Chalechale, "Parallelization of a color DCT watermarking algorithm using a CUDA-based approach," in *Proc. 6th Int. Conf. Comput. Knowl. Eng. (ICCKE)*, Oct. 2016, pp. 100–105.
[35] H. Fan, M. Huang, C. Lai, J. Yu, and W. Xu, "Accelerating DCT-based color image watermarking on GPUs," in *Proc. IEEE Int. Conf. Image Process., Comput. Vis., Pattern Recognit. (IPCV)*, Jan. 2016, pp. 23–26.
[36] H. C. Reeve, III, and J. S. Lim, "Reduction of blocking effect in image coding," in *Proc. ICASSP*, Apr. 1988, pp. 1212–1215.
[37] 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.
[38] W. R. Hamilton, *Elements of Quaternions*. London, U.K.: Longmans Green, 1866.
[39] T. A. Ell and S. J. Sangwine, "Hypercomplex Fourier transforms of color images," *IEEE Trans. Image Process.*, vol. 16, no. 1, pp. 22–35, Jan. 2007.
[40] X.-Y. Wang, W.-Y. Li, H.-Y. Yang, P. Wang, and Y.-W. Li, "Quaternion polar complex exponential transform for invariant color image description," *Appl. Math. Comput.*, vol. 256, pp. 951–967, Apr. 2015.
[41] L. Dagum and R. Menon, "OpenMP: An industry standard API for shared-memory programming," *IEEE Comput. Sci. Eng.*, vol. 5, no. 1, pp. 46–55, Jan. 1998.
[42] C. Chen, K. Li, A. Ouyang, Z. Tang, and K. Li, "Gflink: An in-memory computing architecture on heterogeneous CPU-GPU clusters for big data," in *Proc. 45th Int. Conf. Parallel Process. (ICPP)*, Aug. 2016, pp. 542–551.
[43] K. Li, W. Yang, and K. Li, "Performance analysis and optimization for SpMV on GPU using probabilistic modeling," *IEEE Trans. Parallel Distrib. Syst.*, vol. 26, no. 1, pp. 196–205, Jan. 2015.
[44] W. Yang, K. Li, and K. Li, "A parallel solving method for block-tridiagonal equations on CPUâĂŞGPU heterogeneous computing systems," *J. Supercomput.*, vol. 73, no. 5, pp. 1760–1781, 2017.
[45] L. Wan, K. Li, J. Liu, and K. Li, "Efficient CPU-GPU cooperative computing for solving the subset-sum problem," *Concurrency Comput., Pract. Exper.*, vol. 28, no. 2, pp. 492–516, 2016.
[46] J. Chen *et al.*, "A parallel random forest algorithm for big data in a spark cloud computing environment," *IEEE Trans. Parallel Distrib. Syst.*, vol. 28, no. 4, pp. 919–933, Apr. 2017.
[47] *Nvidia Cuda C Programming Guide*, Nvidia Corp., Santa Clara, CA, USA, 2011.
[48] J. E. Stone, D. Gohara, and G. G. Shi, "OpenCL: A parallel programming standard for heterogeneous computing systems," *Comput. Sci. Eng.*, vol. 12, no. 3, pp. 66–73, 2010.
[49] K. M. Hosny and M. M. Darwish, "A kernel-based method for fast and accurate computation of PHT in polar coordinates," *J. Real-Time Image Process.*, Jul. 2016, pp. 1–13, doi: 10.1007/s11554-016-0622-y.
[50] K. M. Hosny and M. M. Darwish, "Accurate computation of quaternion polar complex exponential transform for color images in different coordinate systems," *J. Electron. Imag.*, vol. 26, no. 2, p. 023021, 2017.
[51] K. M. Hosny and M. M. Darwish, "Highly accurate and numerically stable higher order QPCET moments for color image representation," *Pattern Recognit. Lett.*, vol. 97, pp. 29–36, Oct. 2017.
[52] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli, "Image quality assessment: From error visibility to structural similarity," *IEEE Trans. Image Process.*, vol. 13, no. 4, pp. 600–612, Apr. 2004.
[53] H.-T. Hu and L.-Y. Hsu, "Collective blind image watermarking in DWT-DCT domain with adaptive embedding strength governed by quality metrics," *Multimedia Tools Appl.*, vol. 76, no. 5, pp. 6575–6594, 2017.
[54] Q. Su, Y. Niu, Q. Wang, and G. Sheng, "A blind color image watermarking based on DC component in the spatial domain," *Optik*, vol. 124, no. 23, pp. 6255–6260, 2013.
[55] K. M. Hosny, "Fast computation of accurate Zernike moments," *J. Real-Time Image Process.*, vol. 3, nos. 1–2, pp. 97–107, 2008.

KHALID M. HOSNY (M'04) received the B.Sc., M.Sc., and Ph.D. degrees from Zagazig University, Zagazig, Egypt, in 1988, 1994, and 2000, respectively. From 1997 to 1999, he was a Visiting Scholar with the University of Michigan, Ann Arbor and the University of Cincinnati, Cincinnati, USA. He is currently a Professor of information technology with the Faculty of Computers and Informatics, Zagazig University. He published more than 60 papers in international journals. His research interests include image processing, pattern recognition and computer vision. He is a Senior Member of ACM. He is an editor and scientific reviewer for more than 40 international journals.

MOHAMED M. DARWISH received the B.Sc. (Hons.), M.Sc., and Ph.D. degrees in computer science from the Faculty of Science, Assiut University, Assiut, Egypt. He is currently an Assistant Professor of computer science with the Faculty of Science, Assiut University. His research area is image processing, association rule mining, and medical image analysis.

KENLI LI received the Ph.D. degree in computer science from the Huazhong University of Science and Technology, China, in 2003. He was a Visiting Scholar with the University of Illinois at Urbana Champaign from 2004 to 2005. He is currently a Full Professor of computer science and technology with Hunan University and also the Deputy Director of the National Supercomputing Center in Changsha. He has published more than 130 research papers in international conferences and journals, such as IEEE-TC, IEEE-TPDS, IEEE-TSP, JPDC, ICPP, and CCGrid. His major research areas include parallel computing, high-performance computing, and grid and cloud computing. He is an Outstanding Member of CCF. He serves on the Editorial Board of the IEEE Transactions on Computers.

AHMAD SALAH received the master's degree in computer science from Ain-shams University, Cairo, Egypt, and the Ph.D. degree in computer science from Hunan University, China. His current research interests include parallel computing, computational biology, and algorithms.