Received December 25, 2019, accepted January 9, 2020, date of publication January 13, 2020, date of current version January 21, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.2966251

A Digital Watermarking Encryption Technique Based on FPGA Cloud Accelerator

YANPENG CAO, FENG YU, AND YONGMING TANG
International Joint Research Laboratory of Information Display and Visualization, Southeast University, Nanjing 210096, China

Corresponding author: Yongming Tang (tym@seu.edu.cn)

ABSTRACT Digital watermarking has the properties such as invisibility and anti-aggression, so the digital watermarking technology has been widely used in copyright protection, information hiding. The watermarking technology takes into account the invisibility and robustness of the watermark by controlling the embedding intensity and position of the watermark mainly in the transformation domain. In this paper, discrete cosine transform (DCT) is adopted to transform the given image from spatial domain to frequency domain for adding watermark information. In order to meet the demands of image watermarking batch processing and cloud processing in the future, this paper optimized the DCT algorithm and the data precision, and successfully deployed the designed accelerator kernel on the FPGA cloud platform to speed up the processing of watermarking. The implementation of data processing based on cloud platform is the development trend of big data era. The cloud platform adopted in this paper is based on the OpenCL heterogeneous architecture combining CPU and FPGA. The cloud-based implementation makes digital watermarking application highly extensible, widely shareable, and more secure. The whole system implements a series of complete cloud processes including image decoding, image preprocessing, watermark embedding, and watermarked image encoding. The watermarking algorithm is accelerated by the efficient parallel computing capabilities of FPGA. It can be seen that the result of acceleration is remarkable, providing the state-of-the-art throughput of 1.676 GBps and the highest processing speed of 937 FPS for 800 × 800 sized colorimage.

INDEX TERMS Cloud services, parallel operation, integer DCT, data encryption and watermarking, heterogeneous architecture.

I. INTRODUCTION

Digital watermarking technology is an information hiding technology that indirectly embeds some identification information into a digital carrier without affecting the value of the original carrier [1], [2]. By extracting these watermarks hidden in the carrier, it is possible to confirm the content creator, transmit the secret information, or determine whether the carrier has been tampered with. The basic characteristics of digital watermarking are security, concealment and robustness [3]. The digital watermarking algorithms are further divided into spatial domain algorithms, transform domain algorithms and compressed domain algorithms. Commonly used variation domain algorithms [4] include discrete Fourier transform (DFT) [5], discrete cosine transform (DCT) [6] and discrete wavelet transform (DWT) [7]. The transform domain algorithm usually adds the watermark information to the medium frequency part of the human visual perception [8]. By controlling the intensity and position of the watermark embedding, three is a trade off between the watermark invisibility and robustness, and the watermarked image will realize a better performance.

However, the transform domain algorithm needs to change the image pixel information from the spatial domain to the frequency domain. After adding the watermark information, it needs to be inversely transformed to the spatial domain. Only these two steps have a large number of numerical operations. If the process is only implemented using a CPU, the watermarking process will be time consuming. The problem of low rate is particularly prominent when performing digital watermarking operations on large quantities of images. For example, when using Matlab R2017a to process a color image of 800 × 800 pixel, the watermarking process for a single image requires 3.162s (i7-6700 CPU + GTX745 GPU, 4 × 4DCT). In the digital watermark processing of large-volume images, this disadvantage is magnified,

The associate editor coordinating the review of this manuscript and approving it for publication was Xiaochun Cheng.

and the way of implementing the digital watermarking algorithm only by the CPU does not meet the real-time processing requirements. Therefore, image processing applications are often accelerated with FPGAs which have efficient parallel processing capabilities. Compared with CPU and GPU, FPGA has significant advantages in processing computation-intensive and communication-intensive tasks in terms of speed and power consumption, and can effectively solve the problem of numerical operations in digital watermarking.

Different from the traditional way of developing with FPGA development board or FPGA accelerator card, this paper deploys the design application on the ''cloud'' platform. The concept of ''cloud'' computing was proposed in 2007 [9]. The emerging computing technologies such as distributed processing technology and parallel processing technology together form a ''cloud'' computing system [10]. Nowadays, with the trend of data-centric big data, people's demands for intensive computing and cloud processing are growing. Ultra-large-scale data computing is no longer limited by local computing models with limited capabilities, so more and more applications are adopted on ''cloud'' platform. As a semi-custom circuit in the industry, FPGA has the characteristics of high integration, fast operation and short development cycle. The FPGA integrates a large number of types of programmable logic resources, which can be flexibly designed into complex circuits, and uses parallel operations to implement logic operations. These advantages make FPGAs close to the performance of ASIC technology and become the new favorite of the cloud computing era [11]. The PaaS (Platform-as-a-service) heterogeneous cloud platform composed of FPGA and CPU can create high-performance accelerators for different designs, which will provide a reliable speed guarantee for the development of this design. The FPGA cloud platform used in this design is based on Xilinx Ultrascale+. This kind of FPGA has a large amount of logic and memory resources, strong scalability, which can greatly reduce the delay of data in the calculation and transmission process, improve the design speed [12]. Finally, the design needs to coordinate the management of the processor and the hardware acceleration part. The OpenCL architecture is used to integrate the entire design, realize the efficient interaction between CPU processor and FPGA, and accelerate the image watermarking algorithm.

In this paper, the proposed scheme targets high-throughput and low-latency, cloud applications for robust and blind image watermarking. The most proposed methods concern on the optimization of watermark's quality, on the contrary ignoring the speed of watermarking. Because of this, the already presented designs cannot meet the basic requirements of the actual cloud service. We introduce the integer DCT transform to watermarking system and take several efficient hardware optimization strategies combining the efficient parallel computing ability of FPGA. Our method greatly improves the processing speed of digital watermarking for color image. The main contributions of this paper are summarized as follows:

1. High-throughput and low-latency performance: The proposed method provides a state-of-the-art throughput 1.676 GBps efficient watermark accelerator for every processing element. It runs at 300MHz and has a lowest critical path as 2.758ns. The number of clock cycles of processing single image is 320K. In other words, the accelerator processes 937.45 images per second.
2. Hardware friendly optimization strategies: Unroll and Pipeline the DCT transform processer unit to accelerate the loop computation. In addition, optimizing transfer interface between the Host and the accelerator improves data transfer rate. According to the analyzer, optimized implementation is 32.5 times faster than before.
3. Minimal resource utilization: Prototype hardware implementations for 800 × 800 sized image requires only 4 DSP48E (0.058% used), 3405 numbers of slice FF (0.144%), and 7469 slice LUT (0.632%), thanks to the low complexity of proposed method.

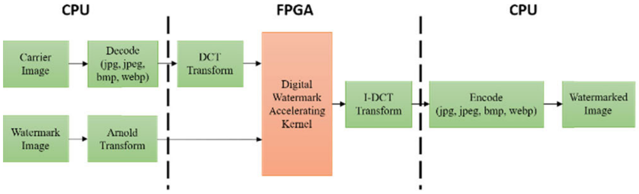


FIGURE 1. System framework of the three implementation parts of digital watermarking allocated in CPU and FPGA respectively.

II. RELATED WORK

A. DIGITAL WATERMARKING

Broadly, watermarking can be divided into two classes: robust and fragile watermarking [22]. Robust watermarks are required to remain in the watermarked image even after it has been manipulated by different attacks whereas the fragile watermarks are designed to be broken easily by image processing operations. In addition, digital watermarking technology can be classified by diverse domains, i.e., spatial domain, transformation domain and compression domain [23].

There are quantities of watermarking schemes [24]–[29] presented to safeguard the digital media, developed in different domains improving robustness and imperceptibility. In the spatial domain, the watermark is embedded in the pixels of the host image signal. Least significant bit (LSB) modification, singular value decomposition (SVD) and difference expansion (DE) based watermarking are more popular in this domain. These approaches employ directly on the pixel values of the test image. The LSB modification [24] was one of the first algorithms proposed for image data hiding. While in the frequency domain, the watermark information is embedded into the spectral components of the host signal. Bianchi et al. proposed the implementations of the discrete Fourier transform (DFT) and the fast Fourier transform (FFT) in the encrypted domain [26]. Dong et al. proposed a novel and feasible watermarking algorithm in the encrypted domain by using DCT [27]. Zheng et al. proposed the implementation of DWT in the encrypted domain, and a method of reducing the data expansion after encrypted domain DWT is given in [28].

A review of literature reveals that transform domain techniques are typically more robust to various attacks as compared with spatial domain techniques. In [30], Shaik et al. compared the different digital watermarking techniques from the basic properties of data hiding (i.e., robustness, security, capacity and so on). Besides, Arora *et al.* [31] evaluated these steganography techniques from other metrics such as perceptual transparency and computation complexity. All of these properties conflict each other. That is an increase in the payload capacity results in a decrease in the imperceptibility of the secret watermark data. Also decrease in the capacity improves the robustness. We have researched plenty of comparative studies to provide a general overview of four methods, which are most common used in watermarking techniques. Table 1 provides the overall performance features in spatial domain (SD) and transform domain (TD) with respect to PSNR (imperceptibility), capacity, robustness, security and complexity.

TABLE 1. Performance evaluation of watermarking techniques.

Domain	SD		TD	
Features	LSB	DE	DCT	DWT
PSNR	High	Moderate	High	Low
Robustness	Low	Low	Moderate	High
Capacity	High	High	Moderate	Low
Security	Low	Low	High	High
Complexity	Low	Low	Moderate	High

Taking the advantages of energy compaction property, robustness of JPEG compression attacks and good performance in perceptual transparency, DCT based watermarking technique is more reliable for most application. What's more, DCT shows a good performance in case of hardware design [32], performing lower cost and higher throughput contrast to DWT, mainly because of lower computational complexity. Hence, we employ DCT transform as the backbone of our digital watermarking implementation.

B. HARDWARE IMPLEMENTATION

Several hardware implementations of reversible watermarking (RW) [32]–[39] are available in the existing literature. The Very Large Scale Integration (VLSI) architecture for a conventional watermarking algorithm in the spatial domain proposed by Gerimella *et al.* [33] might be considered as a noteworthy early work. Mohanty *et al.* [34] put forward a VLSI architecture that could insert invisible or visible watermarks into digital images in the DCT domain. Recently, Das *et al.* [35] give a RW architecture based on difference expansion. In [34] the first low-power watermarking chip was introduced.

Apart from these VLSI architectures, there are a few FPGA-based implementations [32], [36]–[39] in the recent literature. In [36], Basu et al. suggested a hardware implementation of a fragile watermarking system operating in the spatial domain. Their proposed watermarking scheme was imperceptible but fragile against attacks. FPGA-based hardware accelerator for real-time watermarking [32] is performed in the DCT domain. An invisible robust video-watermarking approach that also based on the DCT has been presented in [38], which uses integer arithmetic units and is readily adaptable to the H.264 standard. Watermark embedding using the discrete haar wavelet transform approach is stated in [39]. The main disadvantage of the proposed system is that the corresponding hardware design consumed huge hardware resources despite that the system used only the DWT tool.

C. SECURE DATA ON CLOUD

With the increase of digital images in daily life, it is a trend to process big data on cloud. In mobile cloud computing environment, data processing and data storage can be migrated into the cloud servers. For the maintenance of images, cloud storage is an example [40], [41]. Cloud servers also have been introduced to image watermarking application [42]. By the cloud server a unique watermark is inserted in to the encrypted images before images are sent to user [43]. When an illegal image is found, by the watermark extraction method an unauthorized user can be outlined. Privacy-aware reversible watermarking [44] permits a party to entrust the task of embedding watermarks to a cloud service provider without compromising information privacy. In [44], Chang et al. employ both online and offline content-adaptive predictors to assist watermark decoding for various operational requirements. In addition, Ibtihal *et al.* [45] integrate the homomorphic encryption as a service for outsourced images in the cloud environment.

III. IMPLEMENTATION FRAMEWORK

The digital image watermarking approach proposed by this paper is parallelized by some optimizing strategies, which is propitious to implement on the hardware. Besides, the integer quantization optimizes the efficiency of image transformation avoided mega floating-point operations.

This paper uses Huawei Cloud FP1 platform to design and accelerate the image digital watermarking algorithm. The FPGA cloud platform is based on the OpenCL [13] heterogeneous processing architecture of the x86 CPU and Virtex UltraScale+ VU9P FPGA. The FPGA is connected to the CPU through the PCIe Gen3X16 interface, providing up to 300Gbps high-bandwidth Mesh optical interconnect network between the FPGAs. The server single physical node mounts 8 VU9P FPGA accelerator cards, and hangs 64G Bytes@2133Mhz DDR4 to meet a large number of data streams. The designed FPGA acceleration image will be loaded to the cloud server to complete the cloud deployment of the watermarking application, as seen in Figure 2.

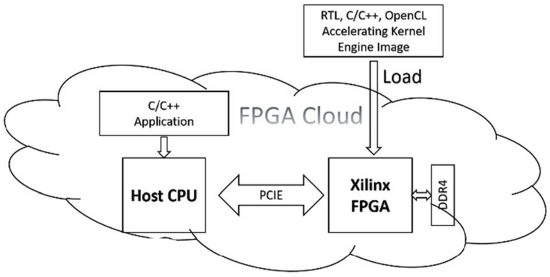


FIGURE 2. Deploy the accelerating application kernel in the FPGA cloud platform.

Combining the advantages of heterogeneous platform processing, we design the processor part mainly responsible for the control and operation logic of the overall program, hand over the time-consuming and complicated numerical operation to the FPGA acceleration.

The overall design framework is as follows: image loading and reading, multiple formats (bmp, jpg, webp, etc.) image encoding and decoding, image preprocessing, watermark image scrambling are placed in the CPU. The image frequency domain transform and the watermark embedding part will be handed over to the FPGA part for accelerating. The OpenCL heterogeneous processing architecture is used for hardware and software collaborative management to realize communication and data transmission between host CPU application and accelerator FPGA program.

Specifically, this paper uses an optimized DCT digital watermarking algorithm. Considering the flexible configurable structure of the FPGA, the algorithm performs numerical operations in an easily accelerated matrix form. Since the color image is composed of three channels of RGB, watermarking for each individual sub-channel will result in a large color shift of the overall image. Therefore, it is considered to convert the RGB image into the YUV space and perform a watermark addition operation on the color layer (U or V channel). The experimental results show that this method has little effect on the overall color perception of the image. In addition, in order to improve the anti-attack capability of the embedded watermark, the Arnold scrambling algorithm is used to process the watermark image to improve the watermark robustness [14].

In this paper, 800 × 800 px color carrier image and 200 × 200 binary watermark image are taken as examples. ImagemagicK tool is used to read and decode images of various formats, and the decoded RGB data is converted into YUV space. The U channel is taken out for 4 × 4 matrix segmentation, divided into 200 × 200 small blocks, and each block 4 × 4 core is subjected to integer cosine transform (ICT) conversion to the frequency domain. The scrambled watermark information is added to the medium frequency part of the human eye's visual perception; the inverse of the integer cosine transform (inverse ICT, iICT) is then used to restore the pixel information to the spatial domain; finally, the encoding is performed, and the encrypted image with the embedded watermark is output. The implementation process is shown in Figure 3.

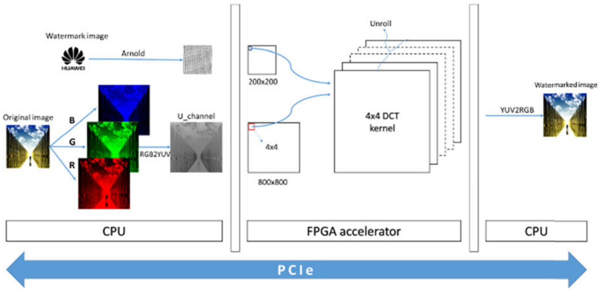


FIGURE 3. Structure diagram of watermarking processing.

IV. THEORETICAL METHOD

A. DCT TRANSFORM ALGORITHM

Discrete Cosine Transform (DCT) algorithm is widely applied in image and video processing field, taking the advantages of the ability of elimination of the pixel coherence in spatial domain [15]. A large number of studies have proved that DCT is a quasi-optimal transformation, with the superior characteristics of real number transform, determined transformation matrix, two-dimensional separable transform and other features. Generally, in DCT frequency domain, its characteristics are as follows. The DCT coefficients are real numbers, which mainly distributed in direct current (DC) component and low-frequency components, while the high-frequency components are few. Therefore, DCT is considered as one of the basic processing modules for images and video processing in many international standards. DCT transform and inverse transform are given respectively in (1) and (2).

$$\begin{aligned} F(u, v) &= \frac{2}{N} \sum_{x=0}^{N-1} \sum_{y=0}^{N-1} f(x, y) \\ &\quad \cdot \cos\left[\frac{\pi}{2N}(2x+1)u\right] \cos\left[\frac{\pi}{2N}(2y+1)v\right] \end{aligned} \tag{1}$$

$$\begin{aligned} f(x, y) &= \frac{1}{N} F(0, 0) + \frac{2}{\sqrt{N}} \sum_{u=1}^{N-1} F(u, 0) \cos \frac{(2x+1)u\pi}{2N} \\ &\quad + \frac{2}{\sqrt{N}} \sum_{v=1}^{N-1} F(u, 0) \cos \frac{(2y+1)v\pi}{2N} \\ &\quad + \frac{2}{N} \sum_{x=0}^{N-1} \sum_{y=0}^{N-1} F(u, v) cos \frac{(2x+1)u\pi}{2N} cos \frac{(2y+1)v\pi}{2N} \end{aligned} \tag{2}$$

It is insensitive for human eyes to perceiving the high frequency information of an image in visual perception. And because of that, adding watermark information by a specific method on the high-frequency components in DCT domain is able to achieve the purpose of the hidden watermark information embedded. In the meanwhile, DC and low frequency components of original image have been reserved, reducing the impact on the visual quality to the largest extent.

The optimization of DCT performance is most important in this design. As far as FPGA is concerned, matrix operation is simple to implement, and can be encapsulated as a standard IP core, thus greatly reducing the design cycle. DCT is an orthogonal linear transformation, which can be formulated in matrix form. The forward and inverse transform of DCT in matric expression is defined below:

$$\begin{cases} DCT : F = C \cdot f \cdot C^T \\ iDCT : f = C^T \cdot F \cdot C \end{cases} \tag{3}$$

where, C is the DCT transformation coefficient matrix.

$$C = \sqrt{\frac{2}{N}} \times \begin{bmatrix} \sqrt{0.5} & \sqrt{0.5} & \cdots & \sqrt{0.5} \\ \cos\frac{1}{2N}\pi & \cos\frac{3}{2N}\pi & \cdots & \cos\frac{2N-1}{2N}\pi \\ \vdots & \vdots & \ddots & \vdots \\ cos\frac{N-1}{2N}\pi & \cos\frac{3\,(N-1)}{2N}\pi & \cdots & \cos\frac{(2N-1)\,(N-1)}{2N}\pi \end{bmatrix} \tag{4}$$

For the two-dimension N × N DCT, the transform used matrix formula requires $2N^3$ multiplications and $2(N-1)N^2$ additions. It can be seen that the computational complexity increases exponentially with the increase of size. Therefore, the simplification based on the block will greatly reduce the numerical computation operations in DCT transform. On the other hand, the energy concentration of watermark is proportional to the size of transform block in DCT frequency domain. Thus, the watermark embedding quality and the anti-attack performance will be better with the increase of N. So that the size of transform block should not be too small. In conclusion, considering both of the computational efficiency and watermark embedding quality, the 4 × 4 block (N=4) is selected as the transform kernel of DCT in our method. Specifically, we will thoroughly discuss this in section VI.A.

B. DCT QUANTITATIVE OPTIMIZATION

Although DCT matrix is all of real numbers, avoiding the complex operation, the problem of calculation accuracy remains. The design of floating-point operation in FPGA is very complicated, and takes up huge system logical resources. In the meanwhile, rounding errors are unavoidable in the floating-point operations, and will accumulate with the increase in the number of operations. In order to ensure a high computational accuracy, there is no doubt that we have to increase the bit width of floating-point operations, which will result in higher computational complexity and lower computational efficiency. Aiming at the above design challenges, DCT transform was optimized to improve the implementation on FPGA. Through mathematical derivation, the equivalent form of integer DCT transform with smaller precision loss was obtained, what is more important is that complex floating-point operation is released from the design on FPGA. The integer cosine transform (ICT) algorithm [16], [17] based on 4 × 4 transform kernel was finally adopted in this paper, which is simplified as follows:

$$\begin{cases} ICT : \left(C_f \cdot f \cdot C_f^T\right) \times E_f \\ iICT : C_i^T \cdot (F \times E_i) \cdot C_i \end{cases} \tag{5}$$

The specific ICT transform expression is given in (6), and the inverse transform expression is given in (7). Thereinto, the forward transformation coefficient matrix C_f and the inverse transformation coefficient matrix C_i are both integer matrices, thus the integer cosine transform will avoid the shortcoming of floating-point operations in DCT.

$$F = \left(\begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & d & -d & -1 \\ 1 & -1 & -1 & 1 \\ d & -1 & 1 & -d \end{bmatrix} \cdot f \cdot \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & d & -d & -1 \\ 1 & -1 & -1 & 1 \\ d & -1 & 1 & -d \end{bmatrix}^T\right) \times \begin{bmatrix} a^2 & ab & a^2 & ab \\ ab & b^2 & ab & b^2 \\ a^2 & ab & a^2 & ab \\ ab & b^2 & ab & b^2 \end{bmatrix} \tag{6}$$

$$f = \begin{bmatrix} 2 & 2 & 2 & 2 \\ 2 & 1 & -1 & -2 \\ 2 & -2 & -2 & 2 \\ 1 & -2 & 2 & -1 \end{bmatrix}^T \cdot \left(F \times \begin{bmatrix} \frac{a^2}{4} & \frac{ab}{4} & & \\ \frac{ab}{4} & \frac{b^2}{4} & & \\ \frac{a^2}{4} & \frac{ab}{4} & & \\ \frac{ab}{4} & \frac{b^2}{4} & & \\ \frac{a^2}{4} & \frac{ab}{4} & \frac{a^2}{4} & \frac{ab}{4} \\ \frac{ab}{4} & \frac{b^2}{4} & \frac{ab}{4} & \frac{b^2}{4} \end{bmatrix}\right) \cdot \begin{bmatrix} 2 & 2 & 2 & 2 \\ 2 & 1 & -1 & -2 \\ 2 & -2 & -2 & 2 \\ 1 & -2 & 2 & -1 \end{bmatrix} \tag{7}$$

where, a $= \frac{1}{2}$, $b = \sqrt{\frac{1}{2}}\cos\left(\frac{\pi}{8}\right)$, $c = \sqrt{\frac{1}{2}}\cos\left(\frac{3\pi}{8}\right)$.

In addition, scale coefficient matrices E_f and E_i in ICT transform can be combined with watermark adding process to simplify calculation, which is represented as W[x]:

$$F_2 = W[F_1 * E_f + kM_{water}] * E_i \tag{8}$$

The watermark information adding process and scale quantification process were carried out at the same time with a few addition and shift operations to realize this whole process. Under this condition, there is no loss of precision even slightly improved, and the calculation complexity is significanty

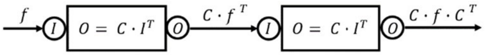


FIGURE 4. Diagram of 2D integer DCT transform simple model.

reduced. Finally, the two-dimensional separable feature of DCT is utilized to transform the dimension of 2D ICT into the same transformation operation with two kernel functions Kernel[I], as defined in (9) and (10).

$$F_f = C_f \cdot f \cdot C_f^T = C_f \cdot \left(C_f \cdot f^T\right)^T \tag{9}$$

$$O = Kernel[I] = C \cdot I^T \tag{10}$$

That is, a 2D ICT transform can be calculated through successively calling the same kernel function twice. Only one ICT transform IP core has to be implemented on FPGA to realize a complete ICT transform. Since the two kernel functions cannot be executed in parallel, the cost in time of time-division multiplexing a same IP core is the same as that of instantiating two IP cores.

The time-division multiplexing of the same IP core through FSM programming on FPGA can greatly improve the utilization rate of logical resources on FPGA and achieve the optimal solution in time and space [18].

C. HLS OPTIMIZATION

For the digital watermarking algorithm proposed in this paper, the hardware execution speed of the whole system is determined by the ICT core IP. Hardware acceleration is particularly important for algorithms with compute-intensive loops. We used Vivado HLS tool to optimize it, iteratively optimized the algorithm from three aspects of operation delay, data throughput and resource occupation through loop flattening, loop pipelining, bitwidth and other optimizations, so as to accelerate the speed of watermarking process [19]. At last, the optimized algorithm was converted into RTL output and packaged the RTL to create the HLS IP. The optimization process is shown in Figure 5.

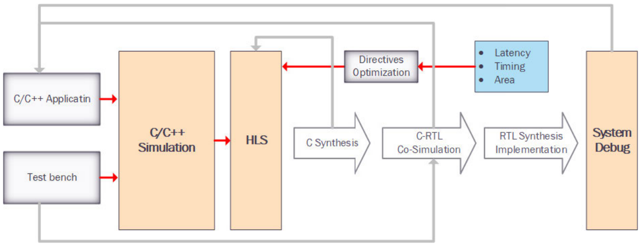


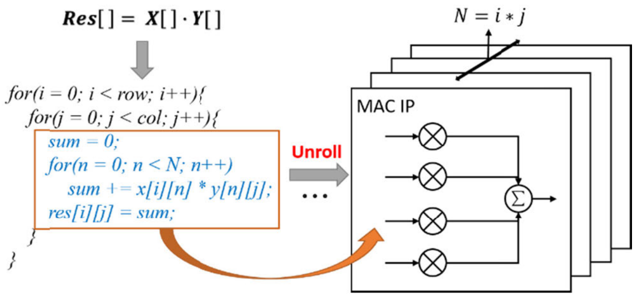
FIGURE 5. The flow diagram of High-Level Synthesis optimization.

Specifically, this paper optimizes the ICT accelerated core from the following three aspects:

1) UNROLL

There are many loop structures in ICT transform, which will greatly increase the overall delay of system execution. Therefore, Unroll optimization is used to flatten the loop structures, which convert the serial operation into parallel operation, appropriately sacrificing space to improve efficiency and speed.

For example, the most basic N × N matrix product operation in ICT transform requires a triple nested for-loop which need N^3 multiplication and addition operation units in serial operation mode. Therefore, it is essential to optimize the loop structure. As seen in Figure 6, the innermost loop is unrolled into parallel structure, and packaged into a multiplier accumulator unit (MAC unit) with N inputs, which constitutes the basic unit in matric operation. The other outer two loops in matrix product are independent from each other in time. Then, N × N basic MAC IP were instantiated at the same time. With the ability of parallel computing on FPGA, one matrix product operation need only one operation cycle.



```
for(i = 0; i < row; i++){
  for(j = 0; j < col; j++){
    sum = 0;
    for(n = 0; n < N; n++)
      sum += x[i][n] * y[n][j];
    res[i][j] = sum;
  }
}
```

FIGURE 6. Diagram of 2D integer DCT transform simple model.

Besides, in the design of image watermarking algorithm, the data blocks are relatively independent, and the basic kernel unit needs to process a large number of data stored in on-chip memory. Therefore, the method of enabling multiple ICT calculation kernel units can be adopted to conduct parallel processing of multiple cores, and the processed data can be temporarily stored in RAM and then transferred to the global memory. In this way, multiple cores performed in parallel will improve the speed of data processing by multiple times; on the other hand, it will take up plenty of hardware logical resources. However, hardware logical resources and memory resources limit the upper limit of the number of processing units. So the number of processing units is determined by the smaller one as defined in (11).

$$\begin{cases} Num1 = Num\_logic/Num\_dct \\ Num2 = Num\_ram/Num\_data \\ Num = min(Num1, Num2) \end{cases} \tag{11}$$

where, *Num_logic* is the total number of logical resources on the chip and *Num_dct* is the number of resources consumed by one processing unit. *Num_ram* is the total number of memory resources on the chip, and *Num_data* represents the number of resources consumed by one unit. *Num1* represents the maximal number, which is available for logical resources to instantiate. Similarly, *Num2* represents memory the maximal number limited by memory resources. Num represents the maximum number of the available units in actual instantiation.

2) PIPELINE

For single 4 × 4 matrix ICT transform, Unroll optimization mentioned above is used to parallel the nested loop. However, for a common digital image, taking 800 × 800 image as an example, 4 × 4 blocked strategy will generate a 200 × 200 repeated block structure, a total of 40,000 blocks independent from each other. These data blocks are independent of each other when they are executed in parallel by taking Unroll optimized strategy. However, the area cost of Unroll is an exponentially increase in hardware resources. Therefore, Pipeline is considered to trade off this problem. Pipelining loop will greatly reduce the operation time delay in the dense loops so that to realize the optimal balance between area and performance [20].

Figure 7 illustrates the processing of the pipeline sequence, where one ICT transform of an independent matrix block includes the following four operations (Read and Write operation are ignored).

- ICT forward transform
- Add watermark
- ICT inverse transform
- Quantify matrix

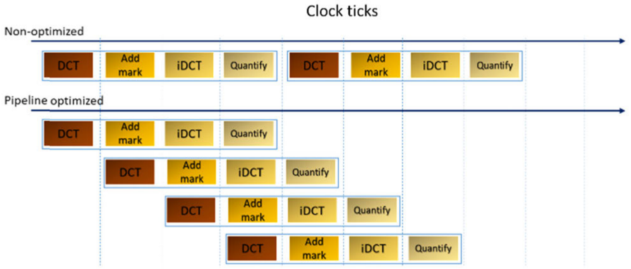


FIGURE 7. **Sequence Diagram of the pipeline timing.**

There are four operation stages formed a 4-states FSM in simple. A common method executing the FSM sequentially requires 4N operation cycles, but the Pipeline optimization only needs N+3 operation cycles in theory, reducing the time delay to 25% of that before the optimization.

In addition, combining loop Unroll and Pipeline will obtain the optimal performance between resource utilization and computing speed. In this paper, Row-Pipeline and Column-Unroll (RP&CU) strategy is adopted to design and optimize, which not only conforms to the row-major order storage feature in digital image, but also trades off the area and performance.

3) RESHAPE THE ARRAYS

The above two approaches optimize the serial loop structure and improve the operation speed. However, a large number of parallel structures impose a huge burden on data throughput. Therefore, it is necessary to optimize the input and output interface of data and the storage form of data in hardware, and even to add extra data buffer to meet the demands of massive burst data bandwidth in parallel computing. It is more flexible for FPGA to design arbitrary bitwidth data types and various custom data interface. By contrast, there are only several fixed data formats (int32, int64, float32, etc.) for designers to use in CPU. For example, pixel data is stored in 8-bit data type for common digital image, but if data in FPGA acceleration core are all stored in 8-bit type, data overflow is bound to occur in multiplication and addition operations. Extending the data bit-width would avoid data overflow problem, such as extending all data to a 16-bit type, surely it would increase the utilization of hardware resources.

Therefore, considering the reasonable data extension is the key to data optimization. Equation (12) provides the constraint of the security data bit-width in the multiplication and addition operations.

$$\begin{cases} Z_{add} = \max(x, y) + 1 \\ Z_{multiply} = x + y \end{cases} \tag{12}$$

In this paper, we adopted the appropriate data bit-width, adjusted the physical implementation of data array, modified the data interface of top function, improved the availability of memory interaction, and optimized the efficiency and precision of data flow interacting between each operation units.

V. HARDWARE OPTIMIZATION

This design adopts Huawei FP1 Cloud platform based on OpenCL heterogeneous platforms. There is a large number of data interactions between FPGA and CPU, so it is necessary to optimize the interactive interface, input and output buffer, data memory block, etc., to improve the throughput bandwidth in the concurrent processing stage.

In the design of OpenCL heterogeneous development framework, we designed the interface in FPGA based on AXI protocol to realize data interaction between FPGA accelerated core and global memory. The CPU host used encapsulated API to access the FPGA kernel and the overall design followed the specification of PCIe bus. This design was developed in SDAccel development environment to optimize the overall hardware acceleration unit, and the specific architecture is shown in Figure 8.

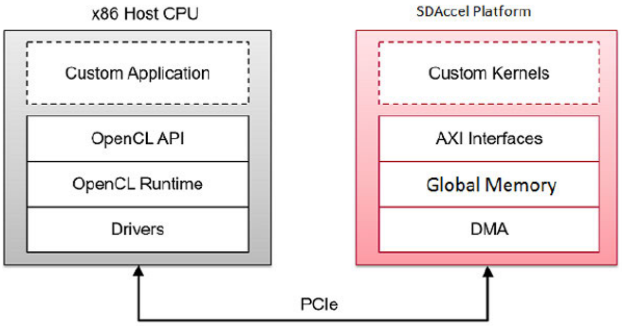


FIGURE 8. **Heterogeneous architecture based on SDAccel development environment.**

In order to improve the efficiency of image watermarking, the hardware optimization in this was divided into the following two parts:

A. BURST TRANSFERS

Transferring data in bursts hides the memory access latency as well as improves bandwidth utilization and efficiency of the memory controller. The size of burst transmission block and data bit-width are the key optimization of this design. FP1 cloud server has dozens of MB on-chip storage resource. By means of PCIe burst transfers, partial data blocks will be loaded into on-chip storage in advance, which can greatly reduce the delay of data transfers. The packet size of burst transfers depends on the interface bandwidth and block length. Using the interface with larger bit-width when the data block length is the same will effectively increase the data throughput.

As mentioned above, the interface between global memory and FPGA accelerated kernel can be configured in SDAccel development environment. Xilinx devices support up to 512 bit-width sizes. In the traditional way, the pixel data after decoded is stored in the memory in the order of row, and the storage space is a 2-dimensional memory block in N × N size.

For single channel cache, it takes 160,000 write operations to store an 800 × 800 pixels image to a buffer. Thus, we optimized the image storage format and data interface. Firstly, the image was partitioned by 4 × 4 block the same as ICT transform kernel. Then each block was arranged into a new array with 128-bit format in order, and it need to recode and re-address the images with the new data format. Finally, 800 × 800 pixels of 8-bit image was organized into 200 × 200 blocks by 128-bit for each block, then written to the memory buffer. The time delay of writing a single image is reduced to 1/16. In this paper, the image data interaction interface between CPU and FPGA is designed uniformly in the above way. Figure 9 shows the corresponding diagram for image re-organization process.

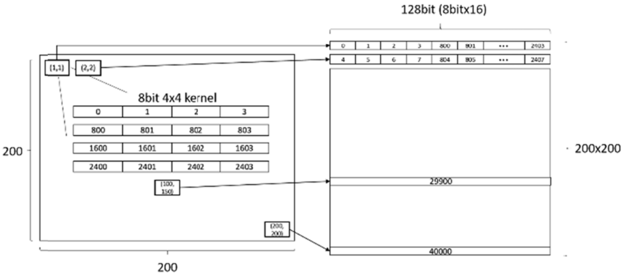


FIGURE 9. Reshape the image matrix.

B. MEMORY INTERFACE OPTIMIZATION

The FPGA part provides 4 independent memory blocks for use, with the highest data transfer bandwidth up to 80GB/s. In the design of FPGA kernel functions, we allocated and enabled different memory blocks to read and write data respectively, and it was divided into two aspects. On one hand, different memory blocks are constrained to different input and output buffers in the host CPU side. On the other hand, multi AXI buses were enabled in the accelerated kernel of FPGA side. In the meanwhile, we designed timing constraints to coordinate different memory blocks with different interface buses. Results demonstrated that it was able to improve the data bandwidth between memory blocks and processing units.

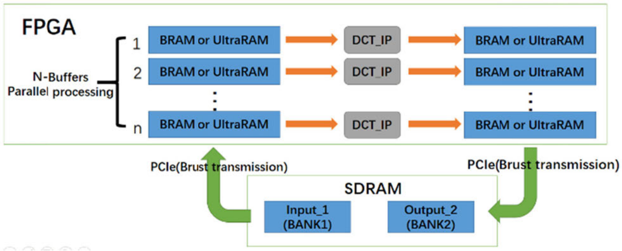


FIGURE 10. Parallel operation and memory optimization.

VI. EXPERIMENTS AND ANALYSES

Datasets. For a fair comparison with other watermarking methods in color image, we use the well-known benchmark dataset: Set5 [46], as shown in Figure 11. Specifically, the image set was released in 2012 by the University of Billie, Bell laboratories, France and consists of five color images of different sizes with natural scenes. Apart from Set5 dataset, we also investigate a larger dataset with a uniform size (800 × 800), as seen in Figure 12. In detail, the set consists of nine images, named Set9. While these images are rendered using artificial patterns and textures and they

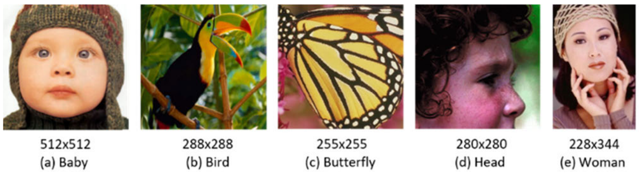


FIGURE 11. Five pictures from the Set5 image dataset.

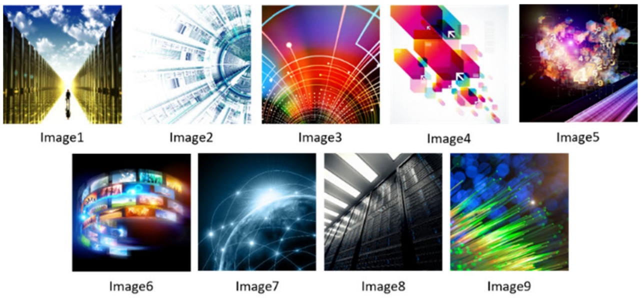


FIGURE 12. Nine pictures from the Set9 image dataset.

TABLE 2. The average values of performance metrics under different size of DCT transform kernel (N from 1 to 10).

Metrics	Kernel Size N									
	1	2	3	**4**	5	6	7	8	9	10
Capacity (bits)	640k	160k	70.76k	**40k**	25.6k	17.69k	12.996k	10k	7.74k	6.4k
Complexity (MAC)	2.56M	5.12M	7.68M	**10.24M**	12.8M	15.39M	17.92M	20.48M	23.04M	25.6M
PSNR (dB)	29.71	32.59	33.99	**35.38**	36.19	36.76	36.86	38.71	38.86	39.26
SSIM	0.8853	0.9240	0.9425	**0.9552**	0.9660	0.9674	0.9694	0.9773	0.9777	0.9802
BER (%)	61.29	1.63	0.29	**0.012**	0.0035	0.0011	0.0001	0	0	0

are all of better visual quality with clear edges. The Set5 is used to evaluate the performance of imperceptibility, and Set9 is used to conduct robustness assessment and analysis different size of DCT transform kernel. Besides, we use three different kinds of watermarks: (a) QR code, (b) Fingerprint, and (c) Flower pattern (see Figure 13).

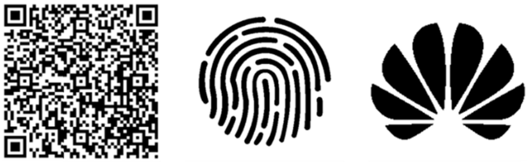

(a) QR-code (b) Fingerprint (c) Flower-pattern

FIGURE 13. Pictures of watermark for test.

Performance Metrics. As mentioned in II.1, there are five basic properties to evaluate a watermarking system: imperceptibility, capacity, robustness, security and complexity. The performance is evaluated by various image quality metrics, like Peak Signal to Noise Ratio (PSNR), Structural Similarity (SSIM) index, Naturalness Image Quality Evaluator (NIQE), Bit Error Rate (BER), and Normalized cross correlation (NC). Rest of the important metrics are well defined in [47]. The mentioned metrics are defined as below:

PSNR: To ensure imperceptibility of the watermarking method and visual inspection of the watermarked image, PSNR is widely accepted as a quantitative metric to measure the quality of the watermarked image. It is formulated as:

$$PSNR = 10log_{10}\left(\frac{MAX_I^2}{MSE}\right) dB, MAX_I = 255 \qquad (13)$$

SSIM: The SSIM is a method for finding the similarity between original image and the watermarked image. It is a perception-based model that considers image degradation as perceived change in structural information. The SSIM measure between two images x and y is represented in Eq. 14:

$$SIM\,(x, y) = \frac{(2\mu_x\mu_y + c_1)(2\sigma_{xy} + c_2)}{(\mu_x^2 + \mu_y^2 + c_1)(\sigma_x^2 + \sigma_y^2 + c_1)} \qquad (14)$$

NIQE: The NIQE [47] is a no-reference image quality score, returned as a nonnegative scalar, which evaluates image quality in a way that is more consistent with human visual perception. Lower values of score reflect better perceptual quality of measured image.

BER: The BER is used to measure the robustness of an algorithm while extracting the watermark data. It indicates the error of extracted bits over total number of embedded bits, and it is given as:

$$BER = \frac{100}{n}\sum_{i=0}^{n-1}\begin{cases}0, W_i = W_i^* \\ 1, W_i \neq W_i^*\end{cases} \qquad (15)$$

NC: The NC is a standard tool for evaluating the degree of similarity between original watermark and extracted watermark. NC between the two images should be very close to unity and is defined as:

$$NC = \frac{\sum_i \sum_j W(i,j) \times W^*(i,j)}{\sum_i \sum_j [W\,(i,j)]^2} \qquad (16)$$

A. INVESTIGATION OF DIFFERENT KERNEL SIZES

There is a sensitive variable N – the kernel size of DCT transform. As the key of DCT based watermarking algorithm, the kernel size N significantly affects the performance of watermarking. To explore a most suitable variable N, we design a group of control experiments with different values of N from 1 to 10, which means the kernel size is 1 × 1, 2 × 2, . . ., 10 × 10, respectively. We test different variable N on image dataset Set9 with three watermarks and evaluate the results with PSNR, SSIM, BER, Capacity, and Complexity, thus we conduct a total of 270 (10 × 9 × 3) experiments. Besides, the capacity (payload) is expressed as the amount of watermark data that it contains. According to the algorithm, there is only one bit of watermark data inserted into single DCT block. Thus, the proposed method can be used to embed $(M/N)^2$ bits information, in total. Where, M and N represent the size of cover image and watermark, respectively. Complexity is described as the total computational cost convoluted in embedding and extracting the watermarks. In general, the computational complexity of matrix multiplication is $O\,(N)^3$. For the two-dimension N × N DCT, the transform requires $2N^3$ multiply accumulate (MAC) operations. Thus, the total number of MAC operations for watermarking an image with a size of M × M is given as below:

$$Complexity : 2\left(\frac{M}{N}\right)^2 \cdot N^3 = 2M^2N \qquad (17)$$

First, we make some statistics for five metrics and calculate the mean value under every size of DCT transform kernel. Specifically, the average values of these metrics are shown in Table 2. Second, we process these metrics to observe the trend more visually by normalizing the minimum and maximum values of each row to (0, 1) in horizontal. This trend is

TABLE 3. Imperceptibility performance evaluation on Set5 and various watermarks. Red and blue indicates the best and the worst performance, respectively.

Carrier Image		Watermark1			Watermark2			Watermark3		
Name	NIQE	PSNR(dB)	SSIM	NIQE	PSNR(dB)	SSIM	NIQE	PSNR(dB)	SSIM	NIQE
Baby	2.5399	35.16	0.9528	2.411	35.15	0.9528	2.412	35.16	0.9528	2.409
Bird	4.2988	35.86	0.9653	3.951	35.87	0.9653	3.947	35.86	0.9653	3.945
Butterfly	5.6540	35.96	0.9885	5.675	35.96	0.9885	5.686	35.97	0.9885	5.629
Head	8.1338	35.35	0.9406	8.018	35.36	0.9406	8.033	35.35	0.9405	8.079
Woman	3.6225	35.20	0.9626	3.306	35.19	0.9625	3.342	35.19	0.9625	3.343

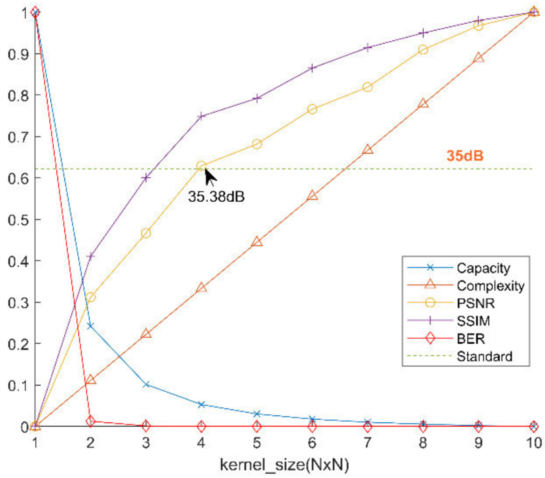


FIGURE 14. Normalization curves for different kernel size.

reflected in Figure 14, it is intuitional that these metrics are conflicted with each other. PSNR and SSIM metrics increase along with the increase of N, and BER drops distinctly, which means that larger N contributes to better performance. Yet, there is a clear turning point when N = 4, and the trend of growth has slowed markedly. In contrast, the computational complexity increases linearly with the raise of N and the capacity reduces exponentially. Therefore, a smaller kernel size will greatly reduce the numerical computation operations in DCT transform and improve the ability of watermark payload.

Indeed, an image with a PSNR of more than 35dB is considered to be high quality and not below 30dB, and for SSIM, the typical value is 0.93 [48]. From all the results, we find the best trade-off between performance and parameter: N = 4, which achieves moderate complexity and capacity, suitable imperceptibility and reliable robustness.

B. IMPERCEPTIBILITY ANALYSIS

From the visual point of view, the digital watermarking method proposed has great imperceptibility owing to the algorithm based on DCT transform domain. We choose the common used benchmark image dataset Set5 (Fig. 11) as the carrier image. The experimental outputs of the embedding and extracting system have been generated using three binary images as the watermark (Fig. 13). The digital watermarking algorithm is implemented on the Huawei cloud platform, and we acquire the watermarked images from the actual hardware simulation. According to the embedding process, the watermarked images are produced as the outputs of the embedding system.

As subjective assessment, three pairs of original carrier images and watermark images are given in Figure 15 for example. Besides, we zoom in on some detail parts of the test images by a scaler of 5. Hardly any differences or visual distortions between the original images and their corresponding watermarked images are found from the perceptual view. Instead of this type of qualitative analysis, this section also focused on some quantitative analysis through image quality metrics: PSNR, SSIM, and NIQE, given in Table 3. Through these metrics, the watermarked image is compared to the original one to judge the dissimilarities between them. It is found in Table 3 that the PSNR value is more than 35 dB and the SSIM value is higher than 0.94 for all of the test images. This indicates a high visual transparency of the mark. For NIQE metric, the values of original and watermarked images are close to unity whatever different watermarks. These results imply that the proposed method provides a high imperceptibility in embedding the watermark.

C. ROBUSTNESS ANALYSIS

Robustness is another performance to assess the quality of a watermarking system. Actually, various types of attacks may occur in transmitting process and therefore the image as well as the watermark embedded into it would be distorted. To demonstrate robustness, we conduct the attack test on the watermarked images based on Stirmark Benchmark 4 [49], and the watermark after the attack is extracted. The robustness of the watermark algorithm is analyzed according to BER and NC metrics. During the test, we have performed eight attacks: White noise, Salt & Pepper noise, Gaussian noise, Sharpening, Compression based DWT (haar), Cropping 10% and 20%, and Rotation 10°. The same as part VI.A, we choose Set9 as the test dataset and compute the average scores in terms of PSNR, BER and NC in Table 4. Then, the proposed method is inspected for various attacks and parts of the results attained are reported in Figure 16.

After these empirical tests, we have found that the BER value is lower than 10% and the NC value is higher than 0.9 for all tests except for rotation attack. Generally, the NC value is accepted if its value is greater than 0.7 [50]. It can

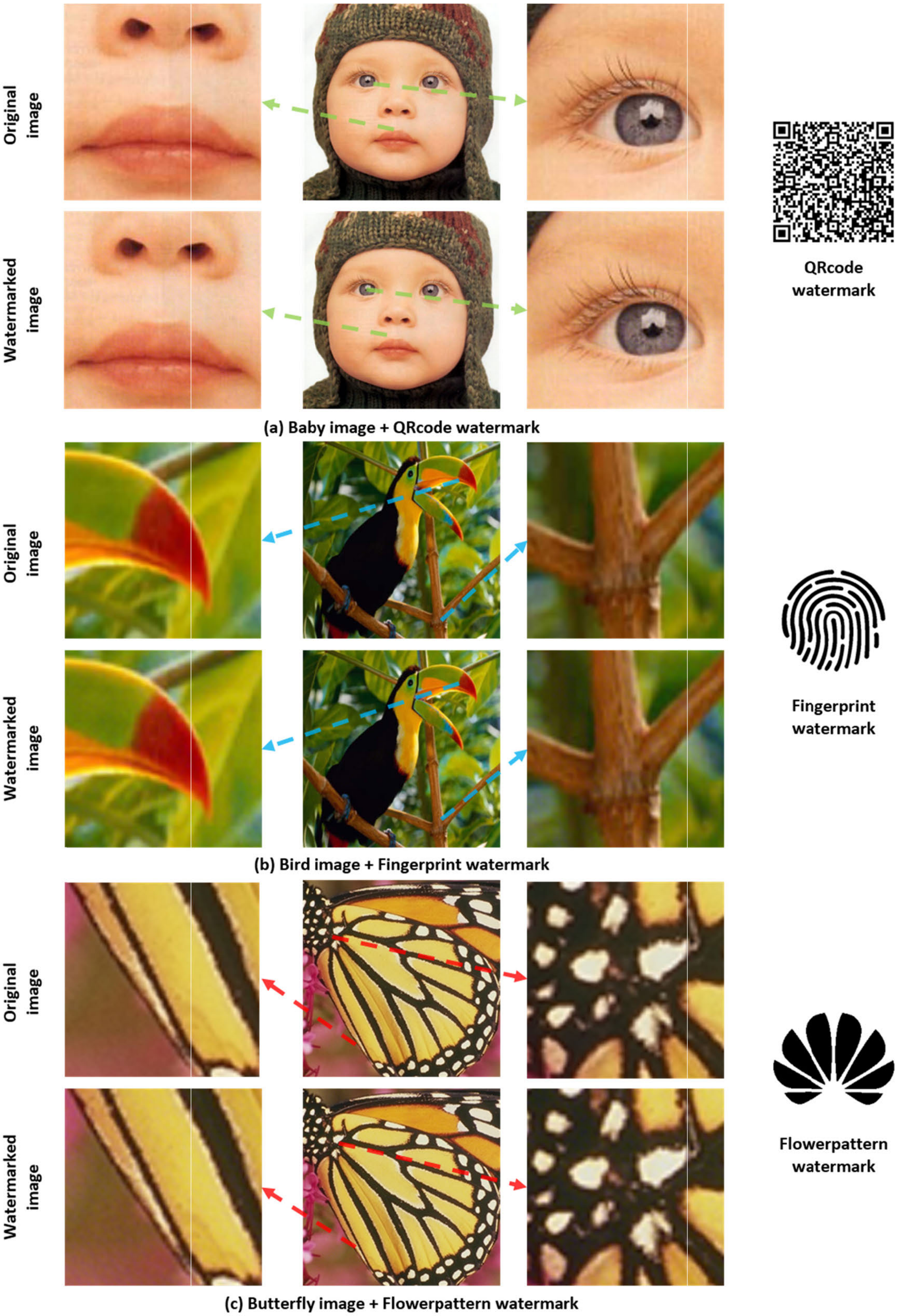
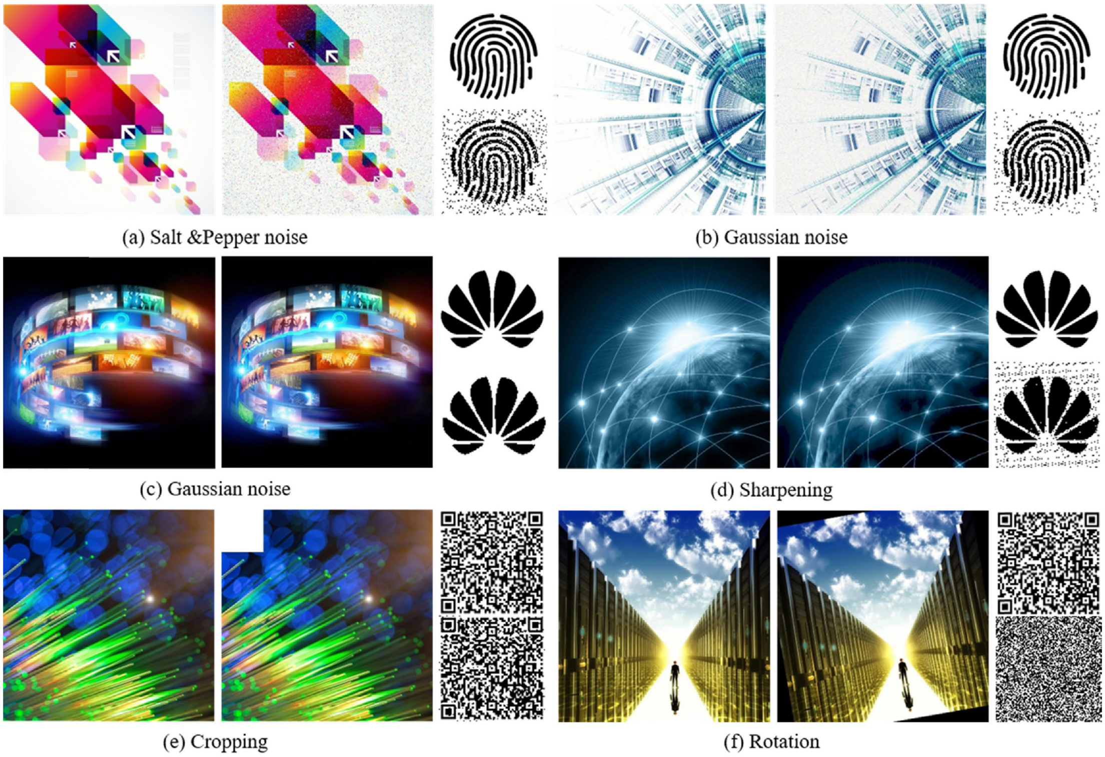


FIGURE 15. Imperceptibility evaluation: The watermarked images of perceptual analysis on Set5 dataset.

TABLE 4. Robustness performance metrics under various attacks. PSNR value shows the different between the original watermarked image and the destructed image after attack. BER < 10% and NC > 0.94 (except for Rotation attack).

Attacks	Watermark1			Watermark2			Watermark3		
	PSNR(dB)	BER(%)	NC	PSNR(dB)	BER(%)	NC	PSNR(dB)	BER(%)	NC
White noise	27.57	0.078	0.9999	27.57	0.010	0.9999	27.57	0.006	1.0000
Salt &Pepper noise	17.07	6.001	0.9421	17.06	6.200	0.9572	17.07	6.104	0.9491
Gaussian noise	20.88	1.870	0.9820	20.88	1.960	0.9866	20.88	1.980	0.9836
Sharpening	29.55	0.012	0.9999	29.55	0.010	0.9999	29.55	0.009	0.9999
Compression(DWT)	27.98	9.203	0.9061	27.98	9.18	0.9189	27.98	9.983	0.9123
Cropping 10%	22.91	0.5133	0.9951	22.91	0.740	0.9950	22.91	0.671	0.9945
Cropping 20%	18.44	2.1333	0.9793	18.44	3.030	0.9792	18.44	2.531	0.9789
Rotation 10°	14.83	50.051	0.5026	13.54	50.16	0.6030	11.84	50.29	0.5368



(a) Salt &Pepper noise

(b) Gaussian noise

(c) Gaussian noise

(d) Sharpening

(e) Cropping

(f) Rotation

FIGURE 16. Robustness evaluation: The watermarks extracted from attacked images on Set9 dataset.

been observed that the proposed method is resilient for most mentioned attacks. However, our DCT-based method resist rotation attacks very well. In summary, the watermarked image performs well in both the noise attack and shear attack (BRE<2%), since the embedding coefficient of the watermark only effects in the medium frequency domain.

D. HARDWARE IMPLEMENTATION ANALYSIS

In this section, we start by presenting the hardware implementation results of the adopted method. The advantages of optimization design based on FPGA cloud platform reflect in processing speed and resource occupation. According to the optimization strategies mentioned before, we optimize the hardware implementation of the watermarking algorithm using HLS tool, and write the corresponding Host file to model the overall hardware implementation architecture. Finally, further optimization analysis is made based on the output report of the cloud platform. The HLS optimization results of the watermarking process are shown in Table 5 and Table 6.

The results show that the timing delay after optimization has a huge speed of improvement. After the optimization operation proposed above, the timing delay is reduced from the original 10.4M to 320K, with a slight sacrifice of resource utilization. The DCT operation integrated on the actual hardware platform using the SDAccel tool, the logical resources are close to the simulation estimation results, meeting the design specifications. It is noted that the proposed architecture is fast and efficient. In the meanwhile, it presents a moderate hardware resources occupation rate.

TABLE 5. Resource utilization of the hardware implementation and corresponding optimized results.

Resources	Available	Used	Percentage	Used	Percentage	Comparison
		Un_Optimized		Optimized		
BRAM_18K	4320	9	0.208%	0	0	Zero
DSP48E	6840	1	0.015%	4	0.058%	×4
FF	2364480	1300	0.055%	3405	0.144%	×2.62
LUT	1182240	4462	0.377%	7469	0.632%	×1.67
URAM	960	0	0	0	0	Unused
Clock = 300MHz						

TABLE 6. Performance comparison between the hardware implementation and corresponding optimized results.

Resources	Un_Optimized	Optimized	Comparison
Critical path (ns)	2.835	2.758	↓ 0.077
Latency (clock cycles)	10400001	320016	↓ 32.5
Capacity (FPS)	28.85	937.45	↑ 32.5
Clock = 300MHz(3.33ns)			

E. COMPARISONS WITH STATE-OF-THE-ARTS

We compare our hardware accelerating implementation with several state-of-the-art methods on max frequency and data throughput respectively. Table 7 shows the comparison results between these proposed methods. According to Table 7, [51] achieved the maximum on the Max Frequency metric of the whole watermarking system, which means that they method would cost a lowest operating latency to process single. It is noticeable that their method only concentrates on small size and gray scaler image with 64 × 64. Although our method only achieves the second best performance next to [51], ours can afford a more powerful ability of processing color image with larger size. Concretely, the data processing capacity is 468.75 (3 × (800 × 800)/(64 × 64)) times of theirs. On the other hand, the highest operating throughput reported in previous work is 800 MBps [32]. However, for our optimization strategy, the maximum operating throughput is 1.676GBps. It is also noticed that the integer optimized strategy causes higher frequency compared to the similar kind of DCT based implementation. Broadly, the proposed method and hardware implementation gives better results.

VII. SUMMARY AND OUTLOOK

Under the trend of big data computing, the algorithmic acceleration processing method based on FPGA cloud platform will play an important role in optimizing the speed of complex algorithm. We implement a digital watermarking algorithm for large-size images on Huawei cloud platform, and pay attention to the robustness performance of watermark. We divide the whole workflow into two parts: software work area and hardware work area. Image reading and writing, decoding and encoding, watermark scrambling operation is processed by CPU, watermarking is accelerated by FPGA, and the heterogeneous processing capability under OpenCL architecture is fully utilized. The quantification of DCT transform and the size of DCT kernel are the key points to achieve the high performance. We explore the most appropriate kernel size in the search space, so as to make a trade off between performance and speed. In addition, we customize an efficient acceleration core for the watermark algorithm based on FPGA. The performance evaluation metric of watermark uses Stirmark tool to test the robustness of the watermarking algorithm under different attacks. Experiment results show that our method performs well in terms of anti-shearing and anti-noise.

The implementation of digital watermarking adopts the classical DCT transform method. For other methods such as Fourier Transform, Wavelet Transform, Walsh Transform, KL transform and other frequency transform, our design framework provided by this paper can be quickly implemented. What's more, the high-efficiency batch processing capability of the proposed system can extend to other image

TABLE 7. Comparison with other state of the art methods. Red and orange indicates the best and the second best performance, respectively.

Method	Year	Method	Device	Max Frequency	Throughput	Size
[39]	2019	DWT (Haar)	Virtex-5	224 MHz	/	512x512 (gray)
[32]	2019	DCT	XCVU440	200 MHz	800 MBps	512x512 (RGB)
[37]	2019	DWT (Lifting)	XC7Z020	130.14 MHz	23.8 MBps	512x512 (gray)
[35]	2018	DE	XCVU095	176.24 MHz	35.3 MBps	512x512 (gray)
[51]	2018	HBS	Virtex-7	445.33 MHz	/	64x64 (gray)
[52]	2017	PC+SVD	Virtex-7	351.46 MHz	/	512x512 (gray)
[53]	2014	FWHT	Virtex-7	259.2 MHz	/	256x256 (gray)
Ours		IDCT	XCVU9P	362.58 MHz	1.676 GBps	800x800 (RGB)

or video processing applications such as image compression [22] and video digital encryption. Our cloud acceleration method has broad application prospects.

ABBREVIATIONS

AXI: Advanced eXtensible Interface;

DCT: Discrete cosine transform;

FSM: finite-state machine;

HLS: High-level synthesis;

ICT: integer discrete cosine transform;

KL: Karhunen-Loève transform;

IP: Intellectual Property CORE, consists of preconfigured logic functions optimized for FPGAs;

RTL: Register-transfer level

REFERENCES

[1] I. J. Cox, M. Miller, J. Bloom, J. Fridrich, and T. Kalker, *Digital Watermarking and Steganography*. San Mateo, CA, USA: Morgan Kaufmann, 2008.

[2] M. A. Nematollahi, C. Vorakulpipat, and H. G. Rosales, *Digital Watermarking: Techniques and Trends*. Springer, 2017.

[3] M. G. N. Mehta, M. Y. Kshirsagar, and M. A. Tankariya, "Digital image watermarking: A review," *Int. J. Sci. Eng. Technol.* vol. 1, no. 2, pp. 169–174, 2012.

[4] V. M. Potdar, S. Han, and E. Chang, "A survey of digital image watermarking techniques," in *Proc. IEEE Int. Conf. Ind. Inform.*, Aug. 2005.

[5] D. Sundararajan, *The Discrete Fourier Transform: Theory, Algorithms and Applications*. Singapore: World Scientific, 2001.

[6] K. R. Rao and P. C. Yip, *Discrete Cosine Transform: Algorithms, Advantages, Applications*. New York, NY, USA: Academic 1990.

[7] D. Sundararajan, *Discrete Wavelet Transform: A Signal Processing Approach*. Hoboken, NJ, USA: Wiley, 2015.

[8] H.-T. Hu, J.-R. Chang, and L.-Y. Hsu, "Robust blind image watermarking by modulating the mean of partly sign-altered DCT coefficients guided by human visual perception," *AEU-Int. J. Electron. Commun.*, vol. 70, no. 10, pp. 1374–1381, Oct. 2016.

[9] S. Baker, "Google and the wisdom of clouds," *Business Week*, vol. 4064, pp. 48–55, Dec. 2007.

[10] D. C. Marinescu, *Cloud Computing: Theory and Practice*. San Mateo, CA, USA: Morgan Kaufmann, 2013.

[11] C. Fei, S. Yi, Z. Yu, W. Yu, H. Franke, X. Chang, and K. Wang, "Enabling FPGAs in the cloud," in *Proc. 11th ACM Conf. Comput. Frontiers*, 2014.

[12] V. Boppana, S. Ahmad, I. K. V. Ganusov, V. Rajagopalan, and R. Wittig, "UltraScale+ MPSoC and FPGA families," in *Proc. Hot Chips Symp.*, 2016.

[13] D. R. Kaeli, P. Mistry, D. Schaa, and D. P. Zhang, *Heterogeneous Computing With OpenCL 2.0*. San Mateo, CA, USA: Morgan Kaufmann, 2015.

[14] Z. Shang, H. Ren, and J. Zhang, "A block location scrambling algorithm of digital image based on Arnold transformation," in *Proc. Int. Conf. Young Comput. Sci.*, 2008.

[15] A. G. Bor and I. Pitas, "Image watermarking using DCT domain constraints," in *Proc. Int. Conf. Image Process.*, 1996.

[16] J. Zhang and A. T. S. Ho, "An efficient digital image-in-image watermarking algorithm using the integer discrete cosine transform (IntDCT)," in *Proc. Conf. Joint Conf. 4th Int. Conf. Inf., Commun. Signal Process.*, 2003.

[17] C.-K. Fong and W.-K. Cham, "LLM integer cosine transform and its fast algorithm," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 22, no. 6, pp. 844–854, Jun. 2012.

[18] A. Barkalov, L. Titarenko, M. Kolopienczyk, K. Mielcarek, and G. Bazydlo, *Logic Synthesis for FPGA-Based Finite State Machines*. Springer, 2016.

[19] R. Nane, V.-M. Sima, C. Pilato, J. Choi, B. Fort, A. Canis, Y. T. Chen, H. Hsiao, S. Brown, F. Ferrandi, J. Anderson, and K. Bertels, "A Survey and Evaluation of FPGA High-Level Synthesis Tools," *IEEE Trans. Comput.-Aided Des. Integr. Circuits Syst.*, vol. 35, no. 10, pp. 1591–1604, Oct. 2016.

[20] L. V. Agostini, I. S. Silva, and S. Bampi, "Pipelined fast 2D DCT architecture for JPEG image compression," in *Proc. Symp. Integr. Circuits Syst. Design*, 2001.

[21] Q. Ji, "Multispectral image compression based on uniting KL transform and wavelet transform," *Hongwai Yu Jiguang Gongcheng/Infrared Laser Eng.*, vol. 45, no. 2, p. 7, Sep. 2019, doi: 10.3788/IRLA201645.0228004.

[22] V. S. Verma and R. K. Jha, "An overview of robust digital image watermarking," *IETE Tech. Rev.*, vol. 32, no. 6, pp. 479–496, Nov. 2015.

[23] L. Tanwar and J. Panda, "Review of different transforms used in digital image watermarking," in *Proc. 2nd IEEE Int. Conf. Power Electron., Intell. Control Energy Syst. (ICPEICES)*, Oct. 2018.

[24] C.-K. Chan and L. Cheng, "Hiding data in images by simple LSB substitution," *Pattern Recognit.*, vol. 37, no. 3, pp. 469–474, Mar. 2004.

[25] D. M. Thodi and J. J. Rodriguez, "Expansion embedding techniques for reversible watermarking," *IEEE Trans. Image Process.*, vol. 16, no. 3, pp. 721–730, Mar. 2007.

[26] T. Bianchi, A. Piva, and M. Barni, "On the implementation of the discrete Fourier transform in the encrypted domain," *IEEE Trans. Inf. Forensics Security*, vol. 4, no. 1, pp. 86–97, Mar. 2009.

[27] J. Dong, J. Li, and Y. Duan, "A robust watermarking algorithm for encrypted medical images based on DCT encrypted domain," in *Proc. Int. Conf. Electron. Sci. Autom. Control*, 2015.

[28] P. Zheng and J. Huang, "Walsh–Hadamard transform in the homomorphic encrypted domain and its application in image watermarking," in *Proc. Int. Workshop Inf. Hiding*. Berlin, Germany: Springer, 2012, pp. 240–254.

[29] A. Furqan and M. Kumar, "Study and analysis of robust DWT-SVD domain based digital image watermarking technique using MATLAB," in *Proc. IEEE Int. Conf. Comput. Intell. Commun. Technol.*, Feb. 2015, pp. 638–644.

[30] R. Amitharaja, A. Shaik, and V. Thanikaise, "Data security through data hiding in images: A review," *J. Artif. Intell.*, vol. 10, no. 1, pp. 1–21, Dec. 2016.

[31] H. Arora, C. Bansal, and S. Dagar, "Comparative study of image steganography techniques," in *Proc. IEEE Int. Conf. Adv. Comput., Commun. Control Netw. (ICACCCN)*, Oct. 2018, pp. 982–985.

[32] H. Ge and J. Sha, "FPGA-based low-complexity high-throughput realtime hardware accelerator for robust watermarking," *J. Real-Time Image Process.*, vol. 16, no. 4, pp. 813–820, Aug. 2019.

[33] A. Garimella, M. V. V. Satyanarayana, and R. S. Kumar, "VLSI implementation of online digital watermarking technique with difference encoding for 8-bit gray scale images," in *Proc. 16th Int. Conf. VLSI Design*, Jan. 2003, pp. 283–288.

[34] S. Mohanty, N. Ranganathan, and K. Balakrishnan, "A dual voltage-frequency VLSI chip for image watermarking in DCT domain," *IEEE Trans. Circuits Syst. II, Exp. Briefs*, vol. 53, no. 5, pp. 394–398, May 2006.

[35] S. Das, R. Maity, and N. P. Maity, "VLSI-based pipeline architecture for reversible image watermarking by difference expansion with high-level synthesis approach," *Circuits Syst. Signal Process.*, vol. 37, no. 4, pp. 1575–1593, Apr. 2018.

[36] A. Basu, T. S. Das, S. K. Sarkar, A. Roy, and N. Islam, "FPGA prototype of visual information hiding," in *Proc. Annu. IEEE India Conf. (INDICON)*, Dec. 2010.

[37] A. Phadikar, G. K. Maity, T.-L. Chiu, and H. Mandal, "FPGA implementation of lifting-based data hiding scheme for efficient quality access control of images," *Circuits Syst. Signal Process.*, vol. 38, no. 2, pp. 847–873, Feb. 2019.

[38] A. M. Joshi, V. Mishra, and R. M. Patrikar, "Design of real-time video watermarking based on integer DCT for H.264 encoder," *Int. J. Electron.*, vol. 102, no. 1, pp. 141–155, Jan. 2015.

[39] M. A. Hajjaji, M. Gafsi, A. Ben Abdelali, and A. Mtibaa, "FPGA implementation of digital images watermarking system based on discrete Haar wavelet transform," *Secur. Commun. Netw.*, vol. 2019, Jan. 2019, Art. no. 1294267.

[40] G. Hu, D. Xiao, T. Xiang, S. Bai, and Y. Zhang, "A compressive sensing based privacy preserving outsourcing of image storage and identity authentication service in cloud," *Inf. Sci.*, vol. 387, pp. 132–145, May 2017.

[41] M. Marwan, A. Kartit, and H. Ouahmane, "A framework to secure medical image storage in cloud computing environment," *J. Electron. Commerce Org.*, vol. 16, no. 1, pp. 1–16, 2018.

[42] A. Y. Alkhamese, W. R. Shabana, and I. M. Hanafy, "Data security in cloud computing using steganography: A review," in *Proc. IEEE Int. Conf. Innov. Trends Comput. Eng. (ITCE)*, Feb. 2019.

[43] I. A. Adamu and B. Souley, "Performance analysis of text and image steganography with RSA algorithm in cloud computing," *Int. J. Softw. Eng. Appl.* , vol. 9, no. 1, pp. 65–76, Jan. 2018.

[44] C.-C. Chang, C.-T. Li, and Y.-Q. Shi, "Privacy-aware reversible watermarking in cloud computing environments," *IEEE Access*, vol. 6, pp. 70720–70733, 2018.

[45] M. Ibtihal, E. O. Driss, and N. Hassan, "Homomorphic encryption as a service for outsourced images in mobile cloud computing environment," *Int. J. Cloud Appl. Comput.*, vol. 7, no. 2, pp. 27–40, Apr. 2017.

[46] M. Bevilacqua, A. Roumy, C. Guillemot, and M. L. Alberi-Morel, "Low-complexity single-image super-resolution based on nonnegative neighbor embedding," in *Proc. 23rd Brit. Mach. Vis. Conf. (BMVC)*, 2012, pp. 135.1–135.10.

[47] A. Mittal, R. Soundararajan, and A. C. Bovik, "Making a 'completely blind' image quality analyzer," *IEEE Signal Process. Lett.*, vol. 22, no. 3, pp. 209–212, Mar. 2013.

[48] H. Yoshikawa, T. Yamaguchi, and H. Uetake, "Objective evaluation of computer-generated hologram image quality," *SPIE Newsroom*, vol. 10, 2016, p. 6331.

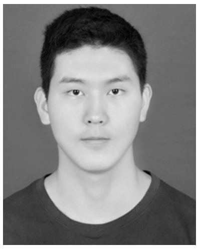
[49] F. Petitcolas, "Watermarking schemes evaluation," *IEEE Signal Process. Mag.*, vol. 17, no. 5, pp. 58–64, Sep. 2000.

[50] H. Guan, Z. Zeng, J. Liu, and S. Zhang, "A novel robust digital image watermarking algorithm based on two-level DCT," in *Proc. Int. Conf. Inf. Sci., Electron. Elect. Eng. (ISEEE)*, Apr. 2014, pp. 1804–1809.

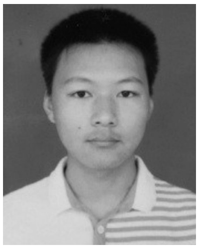
[51] S. Hazra, S. Ghosh, S. De, and H. Rahaman, "FPGA implementation of semi-fragile reversible watermarking by histogram bin shifting in real time," *J. Real-Time Image Process.*, vol. 14, no. 1, pp. 193–221, Jan. 2018.

[52] M. Ranjan Nayak, J. Bag, S. Sarkar, and S. K. Sarkar, "Hardware implementation of a novel water marking algorithm based on phase congruency and singular value decomposition technique," *AEU-Int. J. Electron. Commun.*, vol. 71, pp. 1–8, Jan. 2017.

[53] S. Ghosh, A. Biswas, S. P. Maity, and H. Rahaman, "Design of a low complexity and fast hardware architecture for digital image watermarking in FWHT domain on FPGA," in *Proc. IEEE 5th Int. Symp. Electron. Syst. Design*, Dec. 2014.



YANPENG CAO received the B.E. degree in electronic science and engineer from Southeast University, Nanjing, China, in 2018. He is currently pursuing the M.S. degree with the International Joint Research Laboratory of Information Display and Visualization, Southeast University. His research interests are in the areas of image processing in the encrypted domain, image super-resolution, video enhancement, high-performance computing, and FPGA acceleration.



FENG YU received the B.E. degree in electronic science and engineer from Southeast University, Nanjing, China, in 2018, where he is currently pursuing the master's degree. He majored in digital image processing, algorithm implementation, and high-performance FPGA acceleration. He is currently engaged in high resolution video de-encoder design.

YONGMING TANG received the B.S., M.S., and Ph.D. degrees in physical electronics from Southeast University, Nanjing, China, in 1996, 1998, and 2006, respectively. He is currently a Full Professor with the School of Electronic Science and Engineering, Southeast University, China. He holds more than 30 patents and conference publications. He is engaged in the field of circuit and system and display electronics research. His main research interests include display electronics, video and image processing, and programmable computing.