IEEE Access
Multidisciplinary : Rapid Review : Open Access Journal

Received September 2, 2017, accepted October 5, 2017, date of publication October 17, 2017, date of current version November 7, 2017.

Digital Object Identifier 10.1109/ACCESS.2017.2762399

A Vector-Quantization Compression Circuit With On-Chip Learning Ability for High-Speed Image Sensor

ZUNKAI HUANG[1,2,3], XIANGYU ZHANG[3], LEI CHEN[3], YONGXIN ZHU[1], (Senior Member, IEEE), FENGWEI AN[3], (Member, IEEE), HUI WANG[1], AND SONGLIN FENG[1]

[1]Shanghai Advanced Research Institute, Chinese Academy of Sciences, Shanghai 201210, China
[2]University of Chinese Academy of Sciences, Beijing 100049, China
[3]Hiroshima University, Higashi-Hiroshima 739-8530, Japan

Corresponding authors: Fengwei An (anfengwei@hiroshima-u.ac.jp) and Hui Wang (wanghui@sari.ac.cn)

This work was supported in part by the National Key Research and Development Program of China under Grant 2017YFA0206104, in part by the Shanghai Municipal Science and Technology Commission under Grant 16511108701, in part by the Zhangjiang Administrative Committee under Grant 2016-14, and in part by the China Scholarship Council.

ABSTRACT As the fundamental technology of autonomous vehicles and high-speed tracking, high-speed vision always suffers from the bottlenecks of on-chip bandwidth and storage due to the resource constraints. To improve the resource efficiency, we propose a hardware-efficient image compression circuit based on the vector quantization for a high-speed image sensor. In this circuit, a self-organizing map is implemented for the on-chip learning of codebook to flexibly satisfy the requirements of different applications. To reduce the hardware resources, we present a reconfigurable complete-binary-adder-tree, where the arithmetic units are reused completely. In addition, a mechanism of partial vector-component storage is adapted to make the compression ratio adjustable. Finally, a parallel-elementary-stream design ensures a high processing speed. The proposed circuit has been implemented on the field-programmable gate araray and also applied in a high-speed object tracking system. The experimental results indicate that it achieves an encoding speed of 722 frames/s with 128 weight vectors when working at 79.8 MHz, and the worst tracking error caused by the proposed circuit is merely 9 pixels. These results evince that our proposed circuit can be completely integrated with a high-speed image sensor and used in high-speed vision systems.

INDEX TERMS High-speed image sensor, vector quantization, self-organizing map, field-programmable gate array (FPGA).

I. INTRODUCTION

A. BACKGROUND AND RELATED WORKS

High-speed image sensor has gained significant interest and been widely applied in numerous essential applications such as target tracking [1], [2], optical scientific measurement [3], [4], machine vision [5], [6], etc. With the immense progress of deep submicron CMOS technology, the performance of high- speed CMOS image sensor has been enhanced significantly. However, along with the higher frame-rate and resolution, the insufficient on-chip bandwidth and storage gradually become the bottlenecks that restrict the extensive application of high-speed image sensor. For example, to record the data for a camera with image size of 640 × 480 pixels (VGA size) at 500 frames/s, the required data rate is about 146MB/s, and storing this stream is currently impractical with anything other than dedicated video SRAM. If the camera is integrated with 1 GB of SRAM, only 7 seconds of image data can be recorded before the data are transferred to the hard disk with a slow rate. One of the most effective solutions to overcome the above-mentioned problems is to tightly integrate both image sensor and image compression circuit on a single chip, where the data amount can be reduced to a more manageable level before transmitted.

There is a multitude of standard image/video compression techniques, such as JPEG [7], JPEG2000 [8], H.26X [9], [10], and MPEG [11]. In general, the standard techniques provide excellent compression performances like high image quality and adaptive coding modes. Recently, a number of hardware encoders based on the standard compression techniques have been developed in the previous literature. Sarawadekar and Banerjee [12] proposed an encoder based on embedded block coding with optimized truncation algorithm for JPEG 2000. They devised a new technique named as compact context coding and obtained a high data throughput. The experimental results suggested that their proposed encoder was able to encode the TV sequence at 39 frames/s. In addition, Pastuszak [13] presented an H.264/AVC encoder architecture based on rate-distortion optimization. Their architecture outperformed other designs in terms of compression efficiency, which was even close to that of JM reference software [13]. In [14], Yin *et al.* reported a MPEG-like hardware encoder which was also able to encode the 1080P video sequence. They developed a multi-resolution motion estimation algorithm and thus made the encoder realize a better balance between complexity and performance.

Practically, all of the aforementioned encoders are able to achieve high compression efficiency while keeping low visual quality losses and therefore they can be extensively used in the commercial applications such as TV broadcast, video conference, and digital cinema. However, these schemes must execute complex computations, for instance, discrete cosine transform and Hadamard transform, to acquire a better processing performance. Due to the intrinsic characteristic of high computing complexity, the majority of the hardware encoders based on the state-of-the-art compression standards generally consume significant hardware resources and thus might be not the optimal choice for integration with the high-speed image sensor.

In addition, vector-quantization (VQ) [15], [16] normally has a scalable implementation complexity and potentially high compression ratio. Owing to the inherent parallelism, regular topology, flexibility, and low computing complexity, VQ has been widely used in the multimedia processing applications and communication systems since presented. In the context of the hardware realization of VQ, numerous implementations have been proposed [17]–[20]. Specifically, Fujibayashi *et al.* [17] developed an advanced VQ-based encoder for high-resolution image compression. They introduced a concept of adaptive resolution VQ and also employed a needless calculation elimination method. By this means, the computational cost of their proposed encoder was decreased to 40% or less than the conventional full-search VQ. Similarly, the authors in [19] proposed a novel hardware-oriented SOM (self-organizing map) image compression algorithm. They employed a concept of operator sharing in their architecture to reduce the complexity of each neural cell. Simultaneously, to speed-up the convergence rate, the topological relationship between the neural cells were exploited thoroughly in [19]. Furthermore, Kurdthongmee [20] also presented a hardware-centric algorithm aiming to accelerate the best matching vector searching stage of VQ. He tactfully used the distance as the address of the memory to the store index and thus made the clock cycle for the best matching vector searching process decline to 1 clock.

B. MOTIVATION

In the previous literature of image compression circuits, most authors paid much attention to achieving better visual quality and lower resource usage but often neglected the characteristic of high processing speed. However, for many high-speed vision applications such as high-speed object detecting and tracking, which are regarded as the essential tasks for motion analysis and automatic control, the visual quality of the acquired video sequences are generally not the critical factor restricting the system accuracy. On the contrary, the required fast speed for data processing always seriously restrains their performances. Therefore, developing an image compression circuit with a high encoding speed is quite necessary.

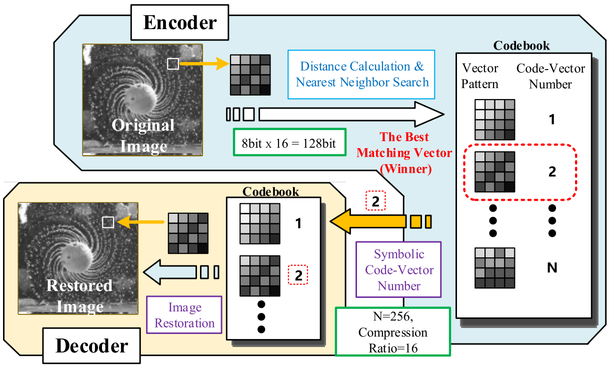

Encoder
Original Image
Distance Calculation & Nearest Neighbor Search
8bit x 16 = 128bit
The Best Matching Vector (Winner)
Codebook
Vector Pattern
Code-Vector Number
1
2
N
Codebook
1
2
2
Symbolic Code-Vector Number
Restored Image
Image Restoration
N=256, Compression Ratio=16
Decoder

FIGURE 1. The basic concept of VQ-based image compression algorithm.

Furthermore, in the majority of the current hardware implementations of VQ-based image compression circuit, the codebook is generated only once by sophisticated algorithms in the software and then loaded to the hardware encoder. Hence, if the application scenes change with time or space, the VQ-based image compression circuit in which the codebook is generated off-line will not be able to meet the demands. One of the most direct and efficient approaches to solving this problem is to integrate the codebook generation part on the hardware encoder so that the encoder can bring the codebook into correspondence with the target senses and applications.

In this paper, we propose a hardware-efficient image compression circuit based on VQ for high-speed image sensor. In this circuit, a self-organizing map (SOM) is implemented for on-chip learning of codebook, and a parallel-elementary-stream (PES) design ensures a high processing speed. Besides, a reconfigurable complete-binary-adder-tree (RCBAT), where the arithmetic units are reused completely, is presented to reduce the hardware usage, and a mechanism of partial vector- component storage (PVCS) is also adopted to make the compression ratio adjustable [21].

C. STRUCTURE

The rest of this paper is organized as follows: Section II briefly explains the fundamental principles of VQ and SOM. Section III describes the proposed circuit for image compression thoroughly. Furthermore, the architecture of a high-speed image sensor integrated with our proposed image compression circuit has been introduced in Section IV. In Section V, the verification system is developed and the experimental results are discussed. Finally, conclusions are given in Section VI.

II. VECTOR-QUANTIZATION-BASED IMAGE COMPRESSION

A. VECTOR QUANTIZATION

VQ is a lossy data compression method, and it has been widely utilized in numerous scientific or industrial applications since presented [22]. As illustrated in Fig. 1, the operations of VQ can be specified as follows: Firstly, the original input image is partitioned into a group of non-overlapping pixel-blocks and each block consists of d pixels. Meanwhile, the pixel values in the individual blocks are arranged to a d-dimensional vector as the input vector. Then, the distances between the input vector and the codebook are calculated. Generally, the codebook is a predetermined set of d-dimensional vectors (code vectors). At the end of the encoding process, the index of the code vector whose distance from the input vector is the nearest is given as the output of the VQ encoder. Since the input vector is more compactly represented by using their index, the data amount requested to be transmitted is reduced obviously. In the decoding process, the image is reconstructed from the compressed data by replacing the indexes with the corresponding code vectors according to the codebook. Formally, VQ can be regarded as a mapping Q from a dd-dimensional Euclidean space R^d to a finite vector set Y in R^d [23] and thus

$$Q : R^d \to Y \tag{1}$$

where $Y = \{y_1, y_2, \ldots, y_N\}$, and $y_i \in R^d$ for $i \in \{1, 2, \ldots, N\}$. Y is the codebook and N is the codebook size.

Among the VQ-based image compression systems, there are mainly three operations, namely codebook generation, image encoding, and image decoding. Practically speaking, codebook generation is the fundamental task in VQ, and the performance of VQ system is highly dependent on the codebook quality. With a large codebook size, the VQ-based encoder can attain better-reconstructed images but will also confront with problems like high computational complexity, time-consuming as well as an increase in memory size. That is to say, there is a tradeoff between the reconstructed accuracy and hardware usage in VQ-based image compression circuit.

The popular codebook generation algorithms in the literature [24]–[26] are the k-mean algorithm and SOM. Especially, SOM holds plenty of advantages such as inherent parallelism, regular topology, and relatively small number of well-defined arithmetic operations involved in its learning algorithm. These features are pretty favorable for hardware implementation of a high-speed image compression algorithm. Therefore, we select SOM as the learning and classification methods in this work.

B. SELF-ORGANIZING MAP

SOM is an unsupervised self-organizing neural network. It defines a nonparametric regression of a set of reference vectors onto the input data [27]. In other words, SOM is a type of nonlinear projection of a probability density function of high dimensional input data onto a two-dimensional array. Since it compresses information while preserving the most important topological and metric relationships of the input data, SOM has been employed to solve problems in a wide variety of application domains. A SOM normally consists of components called nodes or neurons, which are arranged in the form of a two-dimension regular grid. There is a completely-connected network between the input layer and the neuron layer. Each neuron contains a d-dimension vector, W_i, called weight vector, where

$$W_i = \{w_{i1}, w_{i2}, \ldots, w_{id}\}. \tag{2}$$

SOM defines a mapping from the input data space to the grid of neurons. Neurons adapt themselves to the input vector by means of similarity matching between their weight vectors and the input vector. Specifically, the input vectors can be expressed as

$$X = \{x_1, x_2, \ldots, x_d\} \in R^d. \tag{3}$$

The distances to all weight vectors are calculated for each input vector and the neuron whose weight vector is closest to the input vector is regarded as the winner-neuron. The Euclidean distance is one of the most popular ways to measure the distance. It can be defined as

$$D_{Ei} = \sqrt{\sum_{j=1}^{d} \left(x_j - w_{ij}\right)^2}, \quad \text{for } i \in \{1, 2, \ldots, R\} \tag{4}$$

where R is the number of neurons and thus the winner-neuron can be expressed as

$$W_s = arg\,min\,\{D_{Ei}\}, \quad \text{for} i \in \{1, 2, \ldots, R\}. \tag{5}$$

The operation of distance calculation in a high-dimensional space or with a large number of weight vectors is the main source of high computational complexity. In practice, the squared Euclidean distance (SED), (D_E^2), is preferred over D_E in hardware implementation since the root operation has no influence on the distance comparison result but causes additional computational cost.

Once the winner-neuron is determined, the weight vector of the winner-neuron is updated (adapted) by a gradual reduction of the component-wise difference between the input vector and weight vector according to

$$W_s(t+1) = W_s(t) + \alpha(t)\,[X(t) - W_s(t)]. \tag{6}$$

Hence, the update process moves the weight vector of the winner-neuron toward the input vector.

In (6), t is the discrete-time coordinate and $\alpha(t)$ is the learning rate. $W_s(t)$ and $W_s(t+1)$ represent the weight vectors of the winner-neuron before and after updating process, respectively. The winner-neuron searching process in (5) and the weight vector updating process in (6) repeat alternately to facilitate convergence after a multiple of iterations.

III. IMAGE COMPRESSION CIRCUIT WITH ON-CHIP LEARNING

A. THE BASIC ARCHITECTURE

The architecture of the proposed image compression circuit is illustrated in Fig. 2. Basically, the proposed circuit

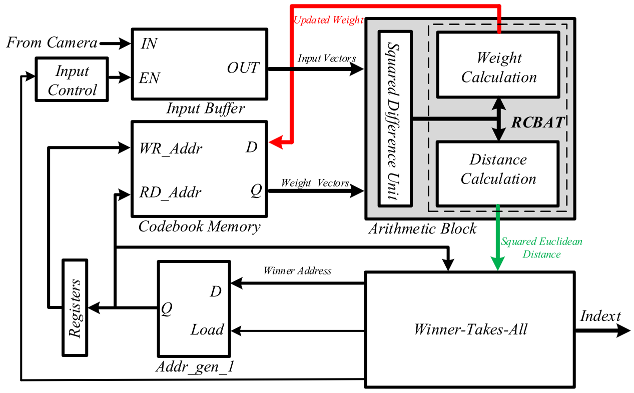


FIGURE 2. Architecture of the proposed image compression circuit.

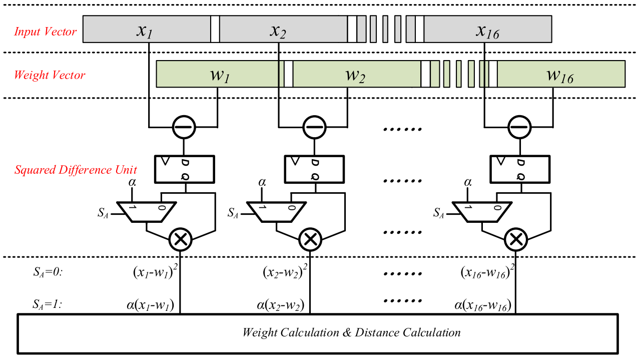


FIGURE 3. Block diagram of the squared difference unit and its related data flow.

is composed of the input buffer, the address generator and memory for the codebook, the arithmetic block, the winner-takes-all (WTA) circuit, and some associate registers. Among these components, the arithmetic block is quite critical. In our design, the arithmetic block not only calculates the distances between the input vector and weight vectors but also generates the new weight vector of the winner-neuron. With the synergy of other parts such as WTA circuit and codebook memory, both the operations of codebook generation and image encoding can be fulfilled by the arithmetic block. As shown in Fig. 2, the arithmetic block can be divided into two parts, namely the squared difference unit (SDU) and the reconfigurable complete-binary-adder-tree (RCBAT). The RCBAT, which reuses the arithmetic logics to reduce hardware resource usage, is able to calculate the distances between the input vector and weight vectors as well as the values of new weight vectors. The diagram of the SDU is shown in Fig. 3. Fig. 4 schematically describes the basic structure of the RCBAT. Furthermore, the WTA circuit, where the winner-neuron is determined and the corresponding index is transmitted, is illustrated in Fig. 5.

B. THE SQUARED DIFFERENCE AND MEMORY BLOCK

As shown in Fig. 3, the elementary SDU consists of a subtractor, a register, a multiplier and an additional multiplexer. In the SDU circuit, we employed a multiplexer to make it be able to switch between the learning mode and the encoding mode. The two input ports of the multiplexers are separately connected to the learning rate α and the outputs of registers. The difference between the input vector and the weight vector are firstly figured out by the subtractor and then latched by the registers. As depicted in Fig. 3, the SDUs are controlled by an independent signal "S_A", and they output either $(x_j - w_{ij})^2$ or α $(x_j - w_{ij})$ which are the basic elements for SED calculating and new weight vector updating, according to the control signal "S_A".

Owing to the fact that different applications usually require different compression ratio, it becomes urgent to develop a more flexible image compression circuit which is able to realize adjustable compression ratio. To meet this demand, we adopt a mechanism of partial vector-component storage (PVCS) to guarantee the flexibility of our proposed circuit for different vector dimensionalities. In PVCS, the d-dimensional input vector "X" and weight vector "W_i" are distributed into p memory blocks, hence each complete vector contains m partial vector-components, where $m = \lceil d/p \rceil$ is the smallest integer not less than d/p. Once the address counter for the vector accessing equals to m, a separation signal "S_{SEP}" is asserted, and this implies an individual vector has been read or written entirely. In terms of our design, the degree of parallelism is set as 16 ($p = 16$), thus a 16-dimensional vector can be processed in every clock. For example, suppose we define the size of the non-overlapped pixel-blocks as 4 × 4 or 8 × 8, the time required for dealing with each vector should be 1 or 4 clock cycles, respectively. Hence, the compression ratio of our proposed encoder can be adjusted according to the requirements of target applications without modifying the circuit architecture.

C. THE RECONFIGURABLE COMPLETE-BINARY-ADDER-TREE

In the arithmetic block, the SDU can only figure out the basic elements, $(x_j$-$w_{ij})^2$ or $\alpha(x_j - w_{ij})$, for calculating distances or new weight vectors. In order to obtain the absolute values of SEDs or new weight vectors, extra operations must be conducted. In most of the previous literature, the authors normally adopted the tree-structured adder to accumulate the SEDs between the input vector and weight vectors. However, to carry out the new weight vectors of the winner-neuron, they have to employ some additional dedicated circuits or external processors. As a consequence, their proposed circuits will inevitably cause various drawbacks, such as large chip area, high power consumption, etc. In our design, we proposed an optimized circuit named RCBAT to solve this problem.

Fig. 4 depicts the basic architecture of the RCBAT circuit. As described previously, the parallelism factor p in SDU is set as 16 in this work, and this means 16 elements will be outputted to the RCBAT per clock. Since each adder in the leaf-side of the adder-tree can summarize 2 results from the

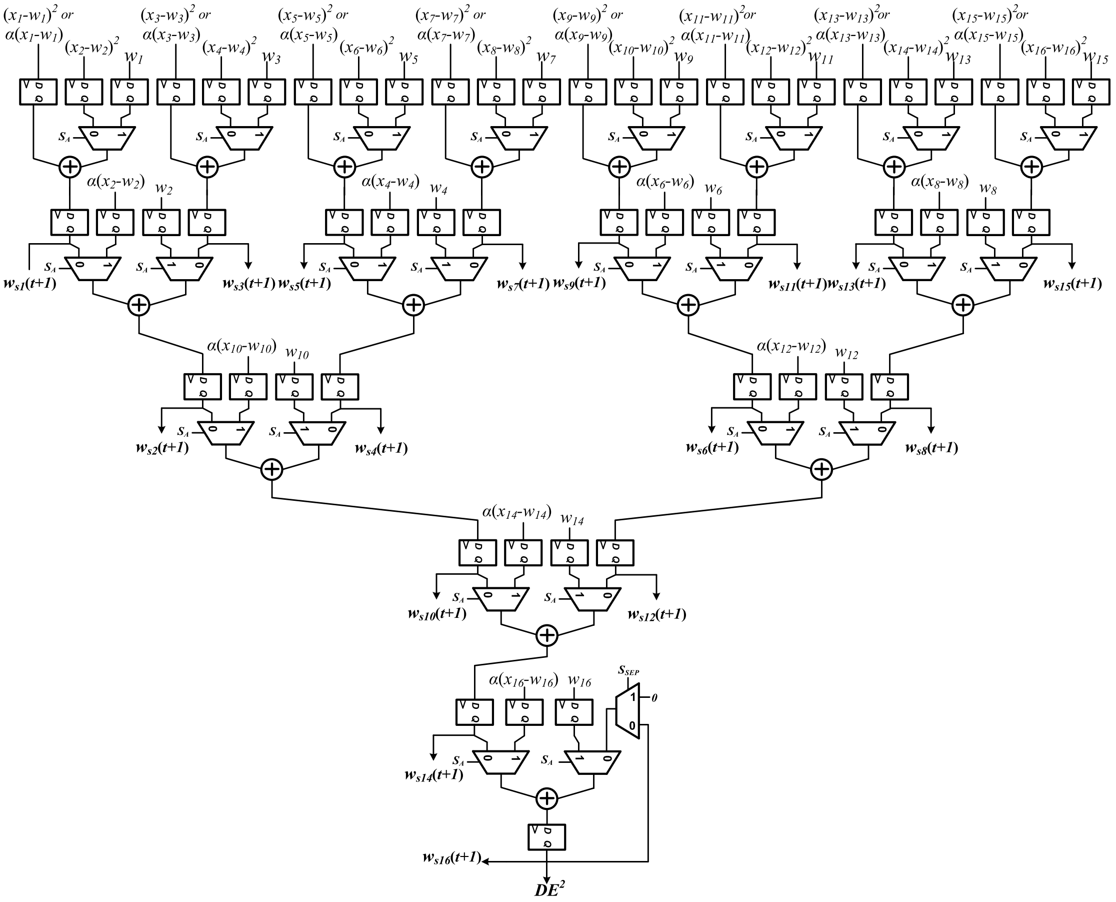


FIGURE 4. Architecture of the reconfigurable complete-binary-adder-tree.

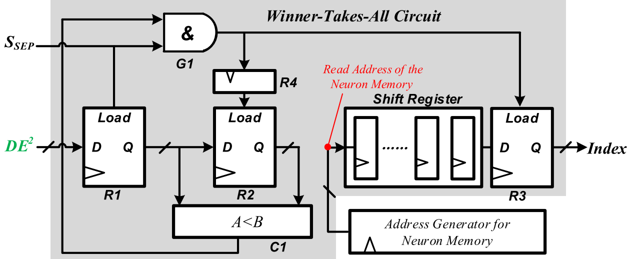


FIGURE 5. Block diagram of the winner-takes-all circuit.

SDU, the corresponding number of adders in the leaf-side will be 8, and the depth of the adder-tree is 4.

The operation principle of the proposed RCBAT can be divided into two modes, namely the encoding mode and the learning mode. In the encoding mode, where "S_A" is set to "0", the values of $(x_j - w_{ij})^2$ are first computed by the SDU and provided to the RCBAT. Then, the 16 squared differences are accumulated in the 4 pipeline stages by the adder-tree. As mentioned before, we adopt a mechanism of PVCS to achieve adjustable compression ratio, and this means we must sum all of the partial SEDs up to get the exact SED. Hence, we add an extra stage under the fourth stage, which is shown in Fig. 4. In this way, the intermediate SEDs can be latched by registers and then totally accumulated by the last adder. The separation signal "S_{SEP}" keeps as "0" until all of the partial vectors have been processed. After the operation of distance calculation, the exact value of SED, "DE^2", which contains m partial SEDs, will be fed into the WTA circuit for local minimum distance searching. Once the winner-neuron is determined, the signal "S_A" is set to "1", and the proposed RCBAT is switched to the learning mode. The beginning address of the winner-neuron is first loaded to the read port of the codebook memory and then the 16-dimensional weight vector of the winner-neuron is read out. During the learning mode, the SDU is configured to compute $\alpha(x_j - w_{ij})$. In the meantime, the RCBAT is transformed to p individual adders to calculate $w_{ij} + \alpha(x_j - w_{ij})$. By this means, the new partial weight vector is obtained. After that, the new partial weight vector is written back via the write-port of the codebook memory. Finally, the entire weight vector of the winner-neuron can be updated after m clocks.

As the SDU and RCBAT can be reconfigured either in learning or encoding mode, all of the arithmetic units can be reused for SED accumulation and weight vector update.

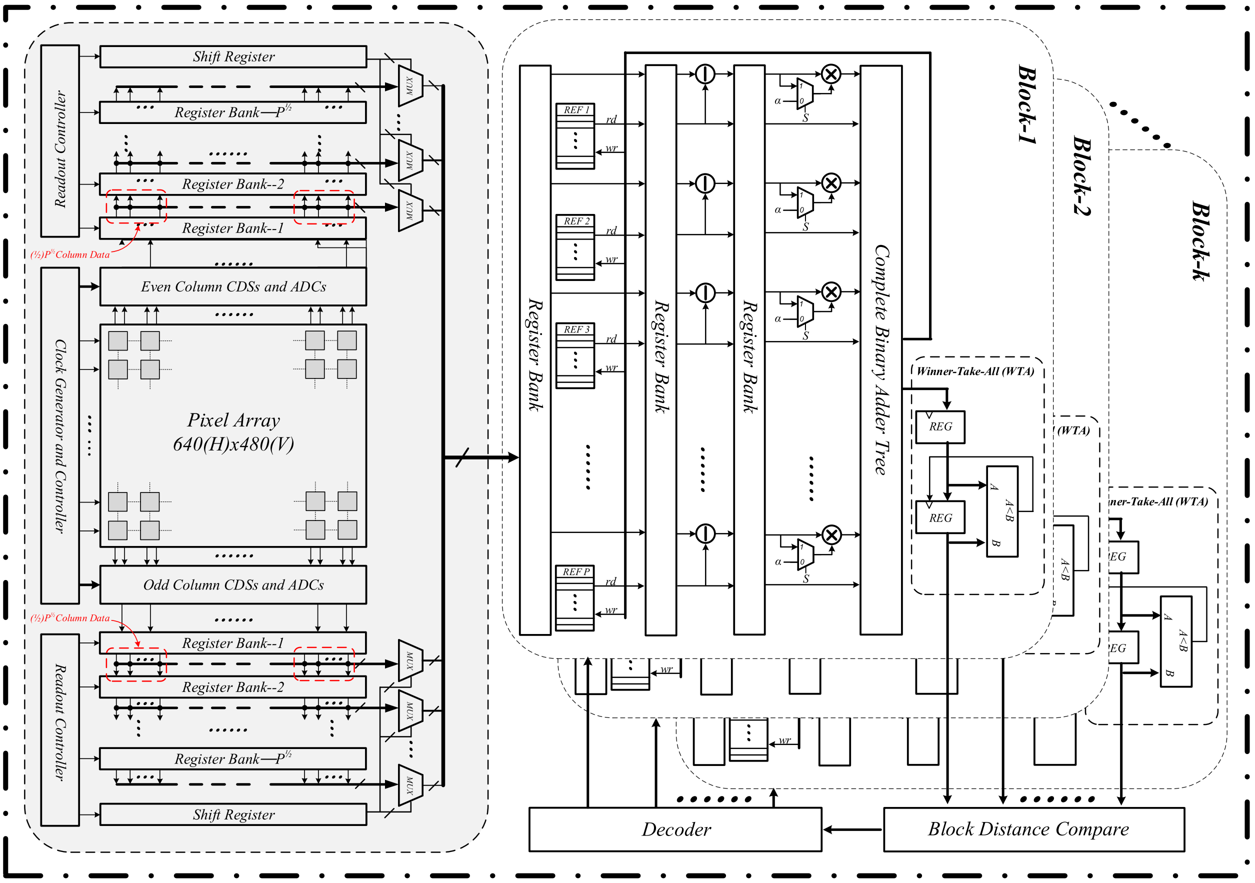


FIGURE 6. A high-speed CMOS image sensor with on-chip image compression circuit.

Hence, the proposed encoder would be capable of generating and modulating the codebook online whereas no additional circuits are required.

D. THE WINNER-TAKES-ALL CIRCUIT

The WTA circuit, as its name implies, acts as an arbitration block to find the smallest SED and determine the winner-neuron. Besides, the address of the winner-neuron, which can be called as index, is also loaded and transmitted by the WTA circuit.

As depicted in Fig. 5, the WTA circuit is basically constructed with one comparator, three main registers with load signal, one *AND* gate and some associate ordinary registers. During the winner-neuron searching phase, the intermediate minimum SED and the corresponding address are temporally stored in *R2* and *R3*. Once the separation signal "S_{SEP}" turns to "1", the newly arrived SED "DE^2" is loaded in *R1* and then compared with the intermediate minimum SED. If "DE^2" is the smaller one, the comparator will output "1" and the values stored in *R2*, as well as *R3*, will be updated accordingly. After all of the weight vectors have been searched, the start address of the codebook memory where the current weight vector is stored will be outputted as an index and transmitted to the receiving terminal.

E. PARALLEL-ELEMENTARY-STREAM DESIGN FOR HIGH-SPEED IMAGE COMPRESSION

The basic principle of the proposed VQ-based encoder for image compression has been introduced and fully explained in the previous sections. It can be concluded that the proposed encoder offers superiorities such as on-chip learning of the codebook, deeply reusing the arithmetic units, and adjustable compression ratio. In this section, based on our proposed encoder, we would like to describe the architecture of a high-speed CMOS image sensor which is integrated with on-chip image compression circuit to clearly show that the proposed encoder is suitable for high-speed image sensor. In our design, a design method named parallel-elementary-stream (PES) is adopted to further improve the processing speed.

Fig. 6 illustrates the general architecture of our design. The entire system mainly contains two parts, namely, the image sensor and the image compression circuit. The image sensor consists of a pixel array, two sets of column correlated-double-sampling circuits (CDS) and analog-to-digital converters (ADC), several register banks, etc. The ADC and CDS arrays are placed at the top and bottom of the pixel array to process the pixels of odd and even columns, respectively. As pixels are accessed in standard row-wise fashion, all pixels in the same row can be read out in parallel. The column-level readout circuits ensure that the CMOS image sensor can work with a high frame-rate while holding a considerable fill factor.

To arrange the pixel values in block form, we serially put v sets of register banks behind the CDSs and ADCs. In this way, v rows of pixels can be provided for reading at the same time. Moreover, v multiplexers, which are also divided into the upper group and the lower group, gather the outputs of each register bank and select the pixel values from left side to right side. The multiplexers are controlled by two sets of shifter registers and each of them selects h pixel values per clock. Hence, a pixel-block with the size of $h \times v$ can be sent to image compression circuit in every clock.

In the image compression circuit, the design method of PES is utilized to increase the processing speed. The elementary block is implemented by the encoder described in Section III. By the PES design, the operation of minimum-distance-searching is performed in parallel. As depicted on the right side of Fig. 6, the codebook is distributed into k blocks. The winner-neurons in all blocks are first found out and the corresponding minimum SEDs are outputted to the block-distance-compare circuit. Then, these k minimum SEDs are further compared and the block in which the winner-neuron exists is determined and selected. Finally, either new weight updating or index transmitting is carried out according to the operation mode.

As introduced in the preceding part, combining image sensing circuit with image compressing circuit into one single chip brings many advantages such as low power, more compact, and free of restriction from on-chip bandwidth, storage, and the I/O bottleneck. This section demonstrates that the proposed image compression circuit can be integrated with high-speed image sensor expediently. This feature may enable our proposed circuit to be implemented in diverse ways.

IV. EXPERIMENTAL RESULTS AND HARDWARE IMPLEMENTATION

A. EXPERIMENTAL ANALYSIS

In VQ-based image compression systems, the quality of the reconstructed image is affected by the learning rate α, the codebook size N, and the number of iteration. In this paper, the learning rate is set as a constant value to simplify the control circuits, which is in a similar way to the previous hardware implementations. Even though we can obtain better image quality with a larger codebook, increasing the codebook size will inevitably cause lower encoding speed and massive silicon area. Furthermore, the number of iteration generally represents some kind of threshold in the learning process. As it has an effect on both the learning time and the final error, we must make a trade-off between speed and accuracy when determining it.

In this section, we have conducted a serial of experiments in regard to the three aforementioned parameters. In order to evaluate the performance of the proposed image compression circuit, the peak signal-to-noise ratio (PSNR) of the reconstructed image is used as a metric [28], which is defined as

$$MSE = \frac{1}{m \times n} \sum\nolimits_{i=0}^{m-1} \sum\nolimits_{j=0}^{n-1} \left[I\left(i,j\right) - K\left(i,j\right) \right]^2 \quad (7)$$

$$PSNR = 10\log_{10} \left(\frac{MAX_I^2}{MSE} \right). \quad (8)$$

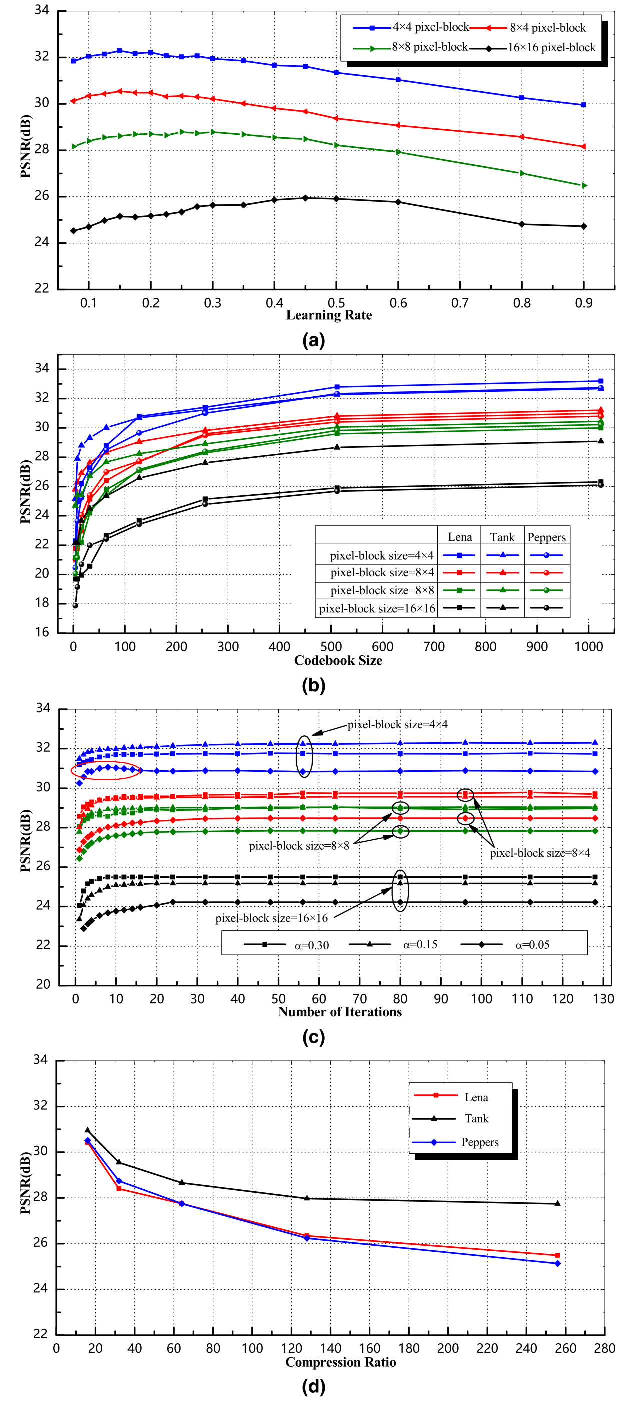
Here, MSE is the abbreviation of mean squared error. In (7), m and n are the total amounts of rows and columns in the images, respectively; $I(i, j)$ and $K(i, j)$ are the pixel values of the original and reconstructed images at (i, j). In (8), MAX_I represents the maximum possible pixel values of the image. For this work, MAX_I equals to 256 since the pixels are represented using 8-bits per sample.

Three grayscale images of 512 × 512 resolution from the USC-SIPI image database, respectively, Lena, Tank, and Peppers, are used for benchmark analysis. The experimental results are depicted in Fig. 7. Considering the trade-off between the hardware resource usage and the visual quality of the reconstructed images, we define the codebook size as 256, which is same with most of the previous VQ-based image compression circuits.

In order to obtain the optimal value of learning rate, we set the number of iterations as 1000, and such configuration is assumed to be practically sufficient for saturation during the learning phase. In addition, several types of pixel-blocks including 4 × 4, 8 × 4, 8 × 8, and 16 × 16 are employed for codebook generation and image encoding. Fig. 7(a) depicts the variations of PSNRs calculated from the reconstructed images with the increase of learning rate. It can be observed that all of the curves seem to be convex functions of the learning rate, and the vertexes of the convex curves are moving to the right side as the size of pixel-block increases. In other words, the optimal learning rate increases when the pixel block becomes large. Furthermore, nearly in all conditions except for the extreme case where a 16 × 16 pixel-block is adopted, the values of PSNR reach the maximum as the learning rate varies around 0.15 to 0.20.

The relationship between the values of PSNR and the size of codebook is illustrated in Fig. 7 (b). Similarly, the number of iterations is set as 1000, and the learning rate is fixed at 0.175. To acquire pervasive results, the values of PSNR for different images with diverse pixel-blocks are calculated and plotted. As shown in Fig. 7 (b), the values of PSNR grow monotonically, but the growth rates only increase slightly with the enlargement of codebook size. For example, the increases in PSNR are no more than 2 dB even if the size of codebook changes from 256 to 512. Hence, with a codebook size of 256, we can obtain a reasonable image quality while keeping a relatively compact memory.

How the number of iterations affects the PSNR is plotted in Fig. 7(c), where the codebook size is fixed at 256 and several kinds of pixel-blocks and learning rate are adopted. Fig. 7(c) manifests that the values of PSNR tend to be nearly steady after the number of iterations exceeds 30.



(a)

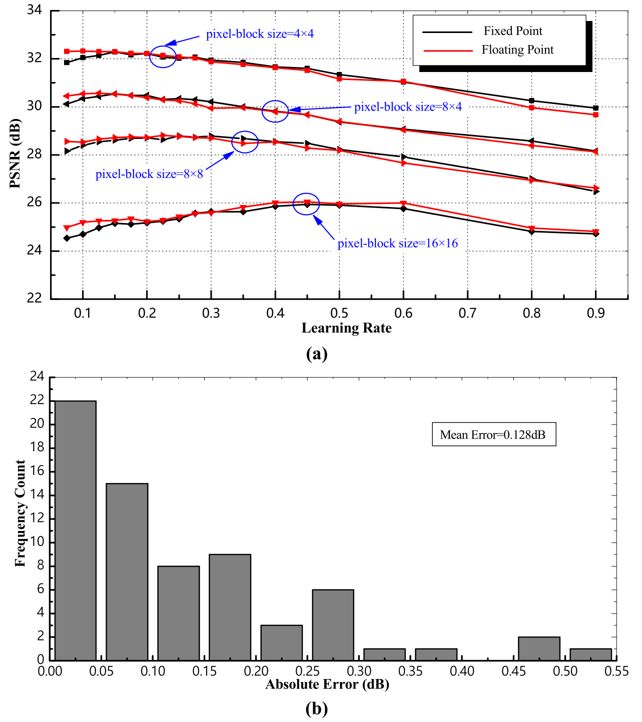
(b)

(c)

(d)

FIGURE 7. Qualities of the reconstructed images under different conditions: (a) PSNR vs. learning rate with different sizes of pixel-block; (b) PSNR vs. codebook size with different sizes of pixel-block for Lena, Tank, and Peppers; (c) PSNR vs. number of iterations with different sizes of pixel-block and learning rate; (d) PSNR vs. compression ratio for Lena, Tank, and Peppers.

In addition, the abnormal fluctuation which is circled by red solid line in Fig. 7(c) arises from the truncation operation of the fixed-point number for hardware implementation. Therefore, the number of iterations was set as 30 in the learning phase, which was demonstrated to be sufficient for saturation.



(a)

(b)

FIGURE 8. (a) Calculated PSNR versus learning rate with various sizes of pixel-block in different data format (b) distribution of the absolute errors between the results obtained by different data format.

Finally, the relationship between the compression ratio and the resulting PSNR of the reconstructed images is studied and illustrated in Fig. 7(d), where the learning rate, codebook size, and learning iterations are respectively set as 0.175, 256, and 30. As explained in Section III, thanks to the functionality of PVCS, the compression ratio of our proposed image compression circuit can be adaptively changed to effectively satisfy the various requirements for different applications.

The accuracy loss caused by the fixed-point calculation in the hardware implementation is also simulated in terms of PSNR. Generally, the accuracy loss mainly steams from the truncation error in the fixed-point calculation. To evaluate the impact of the truncation error, we make a comparison between the fixed-point calculation and the floating point calculation in Fig. 8. The word precision is 8-bit for the pixels and 24-bit for the last adder in Fig. 4. Likewise, the codebook size and number of iterations are respectively set as 256 and 30, and the learning rate varies from 0.075 to 0.9. As shown in Fig. 8(a), even with different data formats, there seems to be merely little PSNR differences caused by the fixed-point calculation. In addition, the truncation noise is expressed as the absolute error of PSNR between the fixed-point calculation and the floating point calculation and then illustrated in Fig. 8 (b). Obviously, the majority of the results are distributed in the area where the absolute errors are relatively small, and the mean error is computed as small as 0.128 dB. This means the truncation error caused by the fixed-point calculation merely brings little quality degeneration to the reconstructed images in our proposed circuit.

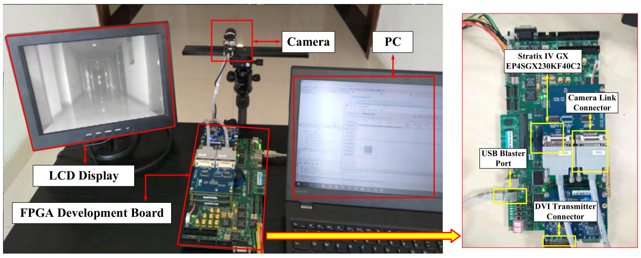


FIGURE 9. A prototype of image compression system based on the proposed circuit.

B. HARDWARE IMPLEMENTATION

The hardware of the proposed image compression circuit was described by Verilog HDL and then implemented on Altera Stratic IV GX-series field-programmable gate array (FPGA). To improve the encoding speed, we define the parallelism degree in the design method of PES as 32, hence the 256 weight vectors are evenly distributed into 32 blocks. Finally, we developed an image compression system based on the proposed circuit. The configuration of the demonstration system is exhibited in Fig. 9.

As shown in Fig. 9, the demo-system is made up of a VGA- sized Camera Link camera with a maximum frame-rate of 500 frames/s, a DE4 FPGA development board, and an LCD display. Furthermore, an independent DIV transmitter-receiver board is inserted to the FPGA development board and the assistant driver is designed and embedded in the FPGA as well. The PC serves as a host to program the configuration files to FPGA device and manage the datasets. The initiatory incoming senses are used as the training data for codebook generation. In this way, a flexible application-oriented codebook can be obtained without any sophisticated algorithm, and the on-chip learning of codebook can be also achieved in the meantime.

TABLE 1. The physical resource utilization of the proposed image compression circuit.

Resources	Used	Available	Utilization
Combinational ALUTs	74368	182400	40%
Memory ALUTs	0	91200	0%
Total registers	62784	N/A	N/A
Total block memory bits	36864	14625792	<1%
DSP block 18-bit elements	0	1288	0%
Total PLLs	1	8	12.5%

The synthesis results of the proposed VQ-based image compression circuit are listed in Table 1. In our design, the multipliers are configured by the look-up tables (LUTs) rather than DSP block elements. For the registers, parts of them are adopted as assistant units to implement the codebook memory and others are used to accomplish operations like data flow modulation, temporary storage, etc. Furthermore, the total memory usage in Table 1 is mainly determined by the codebook size, and it can be further reduced by using a small-sized codebook, especially in applications where the high-quality image is hardly necessary.

To evaluate the processing speed, we have analyzed the required clock cycles to finish the encoding phase and learning phase. The encoding time for one non-overlapped pixel-block is defined as

$$T_{encoding} = \lceil h \times v/16 \rceil \times N/k + 7 \tag{9}$$

where $h \times v$, N, and k are the size of pixel-block, the codebook size and the number of parallelism blocks in PES design, respectively. The number "7" represents the depth of the shift register, which is equal to the sum of 1 stage in the SDU circuit (shown in Fig. 3), 5 stages in the RCBAT circuit (shown in Fig. 4), and 1 stage in the WTA circuit (shown in Fig. 5). Likewise, in the learning phase, updating the weight vector of the winner-neuron additionally takes $\lceil h \times v/16 \rceil + 3$ clock cycles, so the learning time for one non-overlapped pixel-block can be calculated as

$$T_{learning} = \lceil h \times v/16 \rceil \times (N/k + 1) + 10. \tag{10}$$

In essence, the compressing speed of the proposed image compression circuit is determined by the encoding time. For example, if the codebook size N and the number of parallelism blocks k are set as 256 and 32, respectively, the clock cycles required for encoding an 8 × 8-pixel block can be calculated as 39 according to (9). Hence, to encode a frame of grayscale image with VGA size, the proposed image compression circuit will consume 187200 clock cycles, which is about 2.3 ms and equivalent to a maximum frame-rate of 434 frames/s at 79.8 MHz. Furthermore, if the codebook size is decreased from 256 to 128, the corresponding achievable encoding speed will reach 722 frames/s. Finally, we have also synthesized our proposed circuit by the TSMC 0.18μm 1P5M standard CMOS technology. The results indicate that our circuit is able to work at 150 MHz, and this means the encoding speed can reach 1350 frames/s. If we fabricate the circuit with more advanced technologies, we will obtain a much higher compressing speed. Consequently, it is certain that our proposed VQ-based image compression circuit is suitable for the high-speed image sensor.

C. PERFORMANCE DISCUSSION

The comparison results with the previous VQ-based compression circuits are presented in Table 2. To make the data comparable, all of the parameters in Table 2 are calculated under the same condition where the size of the codebook is set as 256. Particularly, the values of PSNR are calculated based on the grayscale Lena with 512 × 512-pixels, and the encoding speed corresponds to the achievable maximum frame-rate for compressing the VGA-sized grayscale video sequences. The results indicate that the maximum frequency of our proposed circuit is higher than that of the previous literature. The better frequency characteristic mainly arises from the inherent simplicity and low-latency of the proposed architecture.

TABLE 2. Comparison results with the previous VQ-based image compression encoders.

Design	[18]	[20]	This Work	
			$k = 16$	$k = 32$
FPGA Family	Virtex II	Virtex IV	Stratic IV	
Devices Used	XC2V6000	XC4VLX200	EP4SGX230	
Compression Ratio (CR)	16 (Fixed)	3 (Fixed)	≥16 (Adjustable)	
Distance Metric	Manhattan	Manhattan	Euclidean	
Frequency (Hz)	71.43M	19.6M	80.1M	79.8M
LUTs	40280	176130	37214	74368
DSP or MULT Blocks	N/A	N/A	0	
PSNR (dB)	31.28	37.10	31.19	
Encoding Speed (Frames/s)	140	92	181 @ CR = 16 235 @ CR = 64 253 @ CR = 256	277 @ CR = 16 434 @ CR = 64 500 @ CR = 256
On-Chip Learning	Yes	No	Yes	

As for the hardware usage, the proposed circuit seems to consume more LUTs than [18] when the parameter k is set as 32. Higher LUTs usage is caused by the large degree of block parallelism, the on-chip learning capability, and the complex distance metric (Euclidean distance in this work). However, if we modulate the parallelism degree to a smaller size, like $k = 16$ shown in Table 2, a lower rate of hardware usage than [18] can be easily reached despite a higher encoding speed than [18] is kept. This mainly benefits from the reusage of arithmetic units in the RCBAT circuit. Furthermore, if we implement part of the multipliers by DSP or MULT blocks, the number of the consumed LUTs will be also reduced definitely. Moreover, Table 2 reveals that our architecture achieves a higher encoding speed than [20], even though [20] just requires fewer clock cycles than this work for the operation of the minimum distance search. This mainly arises from the PES design in our circuit as well as the high latency that introduced by the indispensable register file and other complicated modules in [20]. In addition, it can be seen that owing to the utilization of PVCS, the compression ratio of our proposed circuit is adjustable whereas that of the previous work is fixed. Finally, Table 2 shows that the PSNR in [20] is the best, but this is mainly due to the low compression ratio of [20].

D. SPECIFIC APPLICATION

Generally, compared with the compression standards like JPEG, and MPEG, VQ-based image compression method has an inevitable shortage, namely low PSNR, or relatively lower visual quality [12], [13]. Whereas, the proposed image compression circuit in this work still holds aggressive competitiveness. Above all, the proposed compression circuit is more compact because the circuits based on compression standards usually involve computation intensive algorithms such as discrete cosine transform (DCT) and discrete wavelet transform (DWT). Accordingly, the compact structure makes it practical to integrate the image compression module and CMOS image sensor into a single chip with less hardware expense. More importantly, the proposed compression circuit holds a high encoding speed and thus it is suitable for the data compression task of the high-speed camera.

Compared with the consumer applications, many industrial applications (such as computer vision, motion analysis, and gesture recognition) often require performances of high-speed image capturing and real-time processing. Moreover, since the scenes in most industrial applications normally contain a large amount of invariable background, there must be a high possibility of obtaining a large degree of image compression. In the industrial applications, the proposed compression circuit can show strong advantages over others. In this section, we have applied our proposed circuit in a high-speed object tracking system to demonstrate its practicability. A high-speed camera with 640 × 480 pixels and 500 frames/s is used to continuously capture the images of a fast-moving ball. The codebook is online trained by the initiatory incoming senses, and this ensures its pertinence and effectiveness. For object tracking, the centroid of the moving ball has been extracted. Fig. 10 shows four disparate frames of the original images and reconstructed images with 256 weight vectors. The tracking results of the original frame sequences are used as the comparison benchmark to evaluate the errors caused by image compression.

According to the calculated coordinate, we have depicted the two-dimensional trajectory of the moving target in Fig. 11 (a). Moreover, the quantified coordinate-errors between the reconstructed images and original images over 1000 sample images are calculated and summarized in Fig. 11 (b). The quantified deviations in Fig. 11 (b) are defined as the Euclidean distances between the centroid coordinates in reconstructed images and original images.

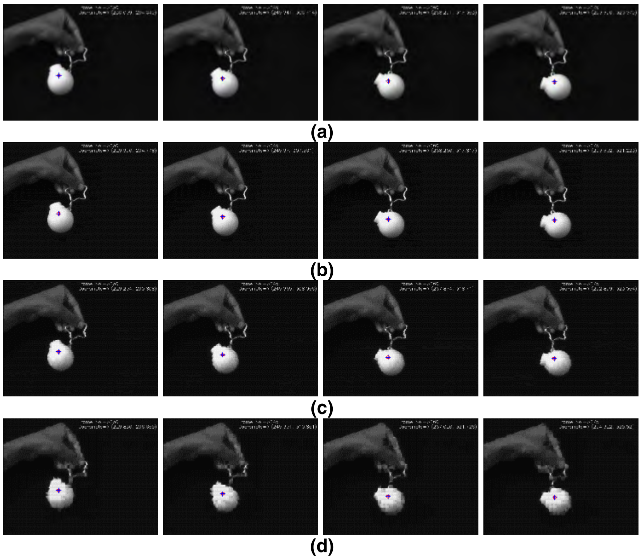
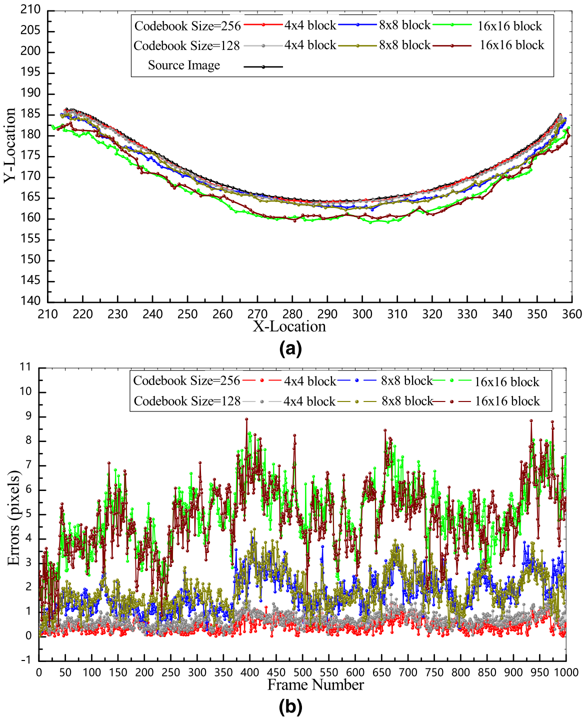

(a)
(b)
(c)
(d)

FIGURE 10. Reconstructed frame sequences with different sizes of pixel-block. (a) Original frame sequences. (b) Pixel-block size is 4× 4. (c) Pixel-block size is 8 × 8. (d) Pixel-block is 16 × 16.



(a)
(b)

FIGURE 11. Experimental results with different kinds of pixel-block and codebook size: (a) Measured two-dimensional trajectory of the moving target; (b) The calculated tracking errors over 1000 frame sequences.

As shown in Fig. 10 and Fig. 11, the visible losses of the reconstructed images only bring about 5-pixel coordinate error even though the compression ratio is nearly 256. Likewise, the trajectories of the reconstructed images merely appear tiny deviations comparing with that of the original images. Furthermore, the worst tracking error can be obtained as about 9-pixels even if the codebook size and pixel block are respectively set as 128 and 16 × 16. More specifically, since the actual dimension of the captured scene in Fig. 10 is about 480 mm × 360 mm, the real maximum error is only about 6.75 mm. This error might be even smaller than the inherent noise of the object tracking algorithm.

V. CONCLUSIONS

In this paper, we present a hardware-efficient image compression circuit for high-speed image sensor to alleviate the bandwidth and storage strain in high-speed vision systems. In this circuit, the on-chip learning of codebook is realized by SOM. Besides, a RCBAT circuit is proposed to reduce the hardware usage by reusing the arithmetic elements. Moreover, benefit from the mechanism of PVCS, modulating the compression ratio according to different requirements is expediently achieved. Simultaneously, the PES design, which holds a high degree of parallelism, ensures a high processing speed. A demonstration system is developed based on the FPGA. The experimental results indicate that the proposed circuit achieves an encoding speed of 722 frames/s with 128 weight vectors when working at 79.8 MHz. Finally, the practicability of the proposed circuit has been also demonstrated by applying it to a high-speed object tracking system, where merely a minor worst tracking error of 9-pixels is caused by our proposed image compression circuit.

REFERENCES

[1] S. Okura *et al.*, "A 3.7 m-pixel 1300-fps CMOS image sensor with 5.0 G-Pixel/s high-speed readout circuit," *IEEE J. Solid-State Circuits*, vol. 50, no. 4, pp. 1016–1024, Apr. 2015.

[2] T. Komuro, I. Ishii, M. Ishikawa, and A. Yoshida, "A digital vision chip specialized for high-speed target tracking," *IEEE Trans. Electron Devices*, vol. 50, no. 1, pp. 191–199, Jan. 2003.

[3] V. Tiwari, M. A. Sutton, and S. R. McNeill, "Assessment of high speed imaging systems for 2D and 3D deformation measurements: Methodology development and validation," *Experim. Mech.*, vol. 47, no. 4, pp. 561–579, Jan. 2007.

[4] D. P. Towers and C. E. Towers, "Cyclic variability measurements of in-cylinder engine flows using high-speed particle image velocimetry," *Meas. Sci. Technol.*, vol. 15, no. 9, pp. 1917–1925, Aug. 2004.

[5] C. Posch *et al.*, "Wide dynamic range, high-speed machine vision with a 2×256 pixel temporal contrast vision sensor," in *Proc. IEEE Int. Symp. Circuits Syst. (ISCAS)*, New Orleans, LA, USA, May 2007, pp. 1196–1199.

[6] L. Lindgren, J. Melander, R. Johansson, and B. Moller, "A multiresolution 100-GOPS 4-Gpixels/s programmable smart vision sensor for multisense imaging," *IEEE J. Solid-State Circuits*, vol. 46, no. 6, pp. 1350–1359, Jun. 2005.

[7] G. K. Wallace, "The JPEG still picture compression standard," *IEEE Trans. Consum. Electron.*, vol. 38, no. 1, pp. 18–34, Feb. 1992.

[8] A. Skodras, C. Christopoulos, and T. Ebrahimi, "The JPEG 2000 still image compression standard," *IEEE Signal Process. Mag.*, vol. 18, no. 5, pp. 36–58, Sep. 2001.

[9] K. Rijkse, "H.263: Video coding for low-bit-rate communication," *IEEE Commun. Mag.*, vol. 34, no. 12, pp. 42–45, Dec. 1996.

[10] Y. W. Chen, K. Chen, S. Y. Yuan, and S. Y. Kuo, "Moving object counting using a tripwire in H. 265/HEVC bitstreams for video surveillance," *IEEE Access*, vol. 4, pp. 2529–2541, May 2016.

[11] D. L. Gall, "MPEG: A video compression standard for multimedia applications," *Commun. ACM*, vol. 34, no. 4, pp. 46–58, Apr. 1991.

[12] K. Sarawadekar and S. Banerjee, "An efficient pass-parallel architecture for embedded block coder in JPEG 2000," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 21, no. 6, pp. 825–836, Jun. 2011.

[13] G. Pastuszak, "Architecture design of the H. 264/AVC encoder based on rate-distortion optimization," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 25, no. 11, pp. 1844–1856, Nov. 2015.

[14] H. Yin, H. Jia, H. Qi, X. Ji, X. Xie, and W. Gao, "A hardware-efficient multi-resolution block matching algorithm and its VLSI architecture for high definition MPEG-like video encoders," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 20, no. 9, pp. 1242–1254, Sep. 2010.

[15] N. M. Nasrabadi and R. A. King, "Image coding using vector quantization: A review," *IEEE Trans. Commun.*, vol. COM-36, no. 8, pp. 957–971, Aug. 1988.

[16] P.-Y. Yang, J.-T. Tsai, and J.-H. Chou, "PCA-based fast search method using PCA-LBG-based VQ codebook for codebook search," *IEEE Access*, vol. 4, pp. 1332–1344, 2016.

[17] M. Fujibayashi *et al.*, "A still-image encoder based on adaptive resolution vector quantization featuring needless calculation elimination architecture," *IEEE J. Solid-State Circuits*, vol. 38, no. 5, pp. 726–733, May 2003.

[18] R. Ramirez-Agundis, G. Gadea-Girones, and R. Colom-Palero, "A hardware design of a massive-parallel, modular NN-based vector quantizer for real-time video coding," *Microprocess. Microsyst.*, vol. 32, no. 1, pp. 33–44, Feb. 2008.

[19] W. Kurdthongmee, "A novel hardware-oriented Kohonen SOM image compression algorithm and its FPGA implementation," *J. Syst. Archit.*, vol. 54, no. 10, pp. 983–994, Oct. 2008.

[20] W. Kurdthongmee, "A hardware centric algorithm for the best matching unit searching stage of the SOM-based quantizer and its FPGA implementation," *J. Real-Time Image Process.*, vol. 12, no. 1, pp. 71–80, Jun. 2016.

[21] X. Zhang, F. An, L. Chen, and H. J. Mattausch, "Reconfigurable VLSI implementation for learning vector quantization with on-chip learning circuit," *Jpn. J. Appl. Phys.*, vol. 55, no. 4S, p. 04EF02, Mar. 2016.

[22] E. Mata, S. Bandeira, M. N. Paulo, W. Lopes, and F. Madeiro, "Accelerating families of *fuzzy K-means* algorithms for vector quantization codebook design," *Sensors*, vol. 16, no. 11, pp. 1963–1982, Nov. 2016.

[23] R. M. Gray, "Vector quantization," *IEEE ASSP Mag.*, vol. 1, no. 2, pp. 4–29, Apr. 1984.

[24] Y. K. Kim and J. B. Ra, "Adaptive learning method in self-organizing map for edge preserving vector quantization," *IEEE Trans. Neural Netw.*, vol. 6, no. 1, pp. 278–280, Jan. 1995.

[25] V. A. Vaishampayan, N. J. A. Sloane, and S. D. Servetto, "Multiple-description vector quantization with lattice codebooks: Design and analysis," *IEEE Trans. Inf. Theory*, vol. 47, no. 5, pp. 1718–1734, Jul. 2011.

[26] M.-H. Horng, "Vector quantization using the firefly algorithm for image compression," *Expert Syst. Appl.*, vol. 39, no. 1, pp. 1078–1091, 2012.

[27] T. Kohonen, "The self-organizing map," *Neurocomputing*, vol. 21, nos. 1–3, pp. 1–6, Nov. 1998.

[28] A. Tanchenko, "Visual-PSNR measure of image quality," *J. Vis. Commun. Image Represent.*, vol. 25, no. 5, pp. 874–878, Jul. 2014.

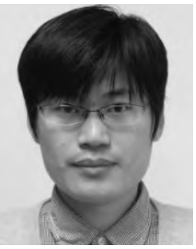
ZUNKAI HUANG received the B.S. degree in electronics science and technology from the School of Electronic Information Engineering, Tianjin University, Tianjin, China, in 2013. He is currently pursuing the Ph.D. degree in microelectronics with the Shanghai Advanced Research Institute, Chinese Academy of Sciences, Shanghai, China. His research interests include in CMOS image sensor chip and system, digital image signal processing circuits, and driving circuits for display panels.

XIANGYU ZHANG received the B.S. degree in electronics science and technology from the School of Electronic Information Engineering, Tianjin University, Tianjin, China, in 2013, and the M.S. degree from Hiroshima University, Higashi-Hiroshima, Japan, in 2016, where she is currently pursuing the Ph.D. degree with the TAOYAKA Program. Her main research is on the hardware development for energy-efficient image recognition algorithms.

LEI CHEN received the B.S. and M.S. degrees from the Qingdao University of Science and Technology, Qingdao, China, in 2006 and 2009, respectively, and the Ph.D. degree from Hiroshima University, Higashi-Hiroshima, Japan, in 2012. She is currently a Post-Doctoral Researcher with the HiSIM Research Center, Hiroshima University. Her research interests include circuit design based on organic thin-film transistors and hardware algorithm design for image processing.

YONGXIN ZHU (M'00–SM'10) received the Ph.D. degree in computer science from the National University of Singapore, Singapore, in 2001. From 2002 to 2005, he was with the National University of Singapore as a Research Fellow. Then, he has been serving as an Associate Professor with the School of Microelectronics, Shanghai Jiao Tong University, Shanghai, China, since 2006. In 2017, he joined the Shanghai Advanced Research Institute, Chinese Academy of Sciences, as a Full Professor. His research interest is in computer architectures, embedded systems, medical electronics, and multimedia.

Dr. Zhu is a Senior Member of China Computer Federation. He is also a Professional Member of the Association for Computing Machinery. He has served over 30 conferences and journals as an Editor, the Program Chair, the Publicity Chair, a Technical Program Committee Member, and a Reviewer.



FENGWEI AN received the B.S. degree from the Qingdao University of Science and Technology, Qingdao, China, in 2006, and the M.S. and Ph.D. degrees from Hiroshima University, Higashi-Hiroshima, Japan, in 2010 and 2013, respectively. He has been an Assistant Professor with the Graduate School of Engineering, Hiroshima University, since 2013, where he is currently an Associate Professor. His research interests include energy-efficient image recognition algorithms and low-power circuits for embedded systems.

HUI WANG received the Ph.D. degree in physics from the Institute of Semiconductors, Chinese Academy of Sciences, Beijing, China, in 2001. He held a post-doctoral position with imec, Belgium, and then as an Associate Professor with Shanghai Jiao Tong University, Shanghai, China. In 2010, he joined the Shanghai Advanced Research Institute, Chinese Academy of Sciences, as a Full Professor in microelectronics. His research interests include high-performance imaging and display panel driving.

SONGLIN FENG received the B.S. degree in physics from Wuhan University, Wuhan, China, in 1983, and the Ph.D. degree in semiconductor physics from Paris University, Paris, France, in 1998.

He was with the Semiconductor Institute, Chinese Academy of Sciences, Beijing, China. In 2001, he transferred to the Shanghai Institute of Microsystem and Information Technology, Chinese Academy of Sciences, where he had served as a Professor and the Director of the Institute. In 2008, he joined the Shanghai Advanced Research Institute, Chinese Academy of Sciences, as the Director of the Institute. He has published over 120 papers in international journals. His current research interests are in the fields of wireless sensor network and microsystem technologies.