Low-cost Hardware Architecture for Integral Image Generation using Word Length Reduction

Junghwan Kim, Jongkil Hyun, and Byungin Moon
School of Electronics Engineering
Kyungpook National University
Daegu, Korea
{jh5746, 26712isjk, bihmoon}@knu.ac.kr

***Abstract*—An integral image is widely used in face detection to calculate feature values at high speed. However, implementing integral images in hardware requires considerable logic and memory resources. This paper proposes a hardware architecture for integral image generation with reduced resource usage by applying the word length reduction method. When implemented in an FPGA, the proposed architecture uses about 83% fewer Slice LUTs than the conventional integral image method. Therefore, the proposed architecture is suitable for low-cost real-time face detection systems.**

Keywords—FPGA; hardware architecture; integral image; face detection

I. INTRODUCTION

In embedded systems, the Haar classifier is generally used for face detection [1]. Accordingly, many studies have been conducted to implement the Haar classifier in hardware. For real-time operation of the Haar classifier, the integral image should be used to minimize the time taken to access the data in memory. However, generating an integral image of the entire image demands large hardware resource, especially as the resolution of the image increases. To solve this hardware resource problem, many studies use architectures that generate an integral image for a sub-window and update the integral image each time the sub-window shifts [2, 3]. Using the sub-window approach, the hardware resources used for storing integral images decrease significantly. Nevertheless, the hardware resources required to generate the integral image are still high because when updating the integral image, values of the leftmost pixels of the integral image before shifting should be subtracted from all other elements of the integral image. Therefore, we propose a real-time integral image generation method and its hardware architecture to reduce hardware resources used in the computation and storage of the integral image.

II. PROPOSED METHOD

The proposed method produces integral images by accumulating pixel values to the right end of the horizontal direction of the input image while the vertical size of integral images is fixed at the height of the sub-window. The proposed method stores the integral image of only the size of the sub-window, and it updates the integral image by simply discarding the oldest values and including newly accumulated right pixel values when the sub-widow shifts. The proposed method requires less computation than conventional methods because it does not subtract the oldest values to update the integral image when the sub-window shifts to the right. However, generating an integral image in this method leads to growth in bit width of the elements in the integral image. To solve this problem, we apply the word length reduction method [4] to each element in the integral image. This is the first study to apply both the sub-window approach and the word length reduction method to the integral image.

The word length reduction method allows the bit width of each element in the integral image to be limited by the maximum sum of the rectangular area used in the calculation of the feature value. This means that even if the integral image reflects the entire horizontal width of the image, the element size of the integral image remains small. Applying the word length reduction method, the cutoff value of the integral image is determined by the maximum sum of the rectangular area used in the Haar classifier. Then, when the element of the integral image is calculated, the cutoff value is subtracted if the accumulated value exceeds the cutoff value. The feature value calculation of the word length reduction method is carried out in a similar way to the traditional feature value calculation process. However, when the sum of the rectangular area is negative or greater than the cutoff value, the cutoff value is added or subtracted as a correction. This requires additional operations to be performed in the feature value calculation but has less negative impact than positive effects as the correction cab be carried out simply by comparison and compensation.

III. PROPOSED ARCHITECTURE

Fig. 1 illustrates the proposed integral image generator. The proposed integral image generator consists of a pixel buffer, a vertical adder tree, adders for the horizontal sum, and an integral image buffer. The pixel buffer consists of 19 line buffers for storing input pixel values. Twenty vertical pixel values, which consists of the current input pixel value and 19 outputs of the pixel buffer, are sent to the vertical adder tree. The vertical adder tree then performs a cumulative sum operation on the 20 vertical pixels to produce 1 × 20 column sum data. The column sum data are summed horizontally with the previous 1 × 20 vertical integral image data in each row to

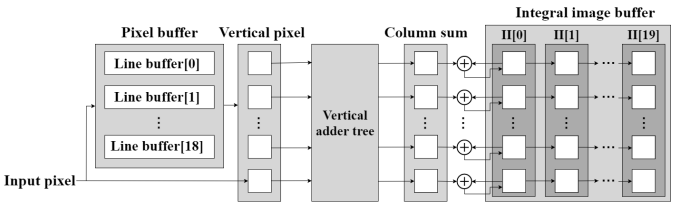

Fig. 1. Proposed integral image generator

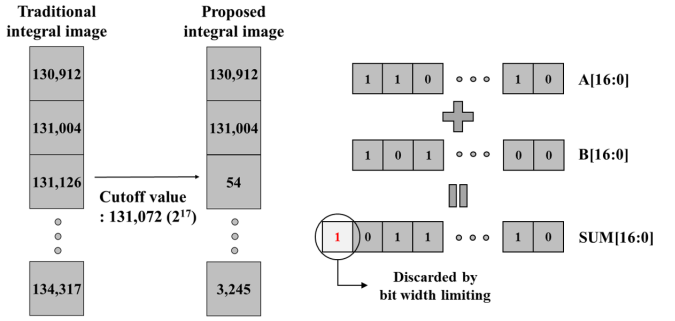

Fig. 2. Word length reduction: (a) cutoff process and (b) proposed cutoff method using bit width limiting

produce a new vertical integral image reflecting the previous values. The vertical integral images are stored sequentially in the integral image buffer to form a 20 × 20 integral image. Each time a new vertical integral image is generated, the data within the integral image buffer are shifted to form a new 20 × 20 integral image for the new sub-window. At this time, the oldest vertical integral image is discarded from the integral image buffer.

Fig. 2 shows an example of the word length reduction in the proposed architecture. As shown in Fig. 2(a), if the sum value is greater than the cutoff value during the integral image generation, the cutoff value is subtracted reflecting the word length reduction method. In this paper, we limit the bit width of the element to remove the additional calculation to subtract the cutoff value, so that the architecture can perform the word length reduction more efficiently. In our architecture, which targets the sub-window size of 20 × 20 and uses grayscale pixels, the maximum sum of the rectangular area is 255 × 20 × 20, and so expressed in 17 bits. For this reason, the proposed architecture limits the bit width of each element in the integral image to 17 bits. Carry outputs from 17 bits in the process of generating the integral image are discarded as shown in Fig. 2(b). The proposed architecture also applies the bit width of 17 bits to the sum of the rectangular area for feature value calculation, thus ignoring the overflow that produces results with negative values or exceeding the cutoff value. This makes the proposed architecture produce the correct feature value without any comparison or compensation.

IV. Experimental Result

Table 1 presents a comparison of the arithmetic units used in our architecture and the architectures of [2] and [3]. All architectures have been designed based on the sub-window of 20 × 20. Unlike the architectures of [2] and [3], the proposed architecture does not require any subtractor to generate the integral image. Thus, the proposed architecture uses fewer arithmetic units than traditional methods.

TABLE I. Comparison of Arithmetic Units for Integral Image Generation

Resources	Architecture used in [2]	Architecture used in [3]	Proposed architecture
Arithmetic Units	20+1 adders, 20×20+20 subtractors	35+20 adders, 20×20 subtractors	35+20 adders

TABLE II. Comparsion of Hardware Utilization

Hardware Utilization	Architecture used in [3]	Proposed architecture
Slice LUTs	7,834	1,300
Slice Registers	7,498	7,510
BRAMs (18 Kbits)	19	19

Table 2 presents the hardware utilization of the proposed architecture and the architecture of [3] when they are implemented on the Xilinx FPGA XC7Z045 FFG900-2. The proposed architecture has greatly reduced the number of Slice LUTs compared with the architecture of [3], while keeping other hardware utilizations almost the same.

V. Conclusion

In this paper, we proposed an integral image generation architecture with reduced hardware resource usage. The proposed architecture applies the word length reduction method to integral image generation based on sliding sub-windows, to reduce logic and memory resources used to create and store integral images. Because of these advantages, the proposed architecture can be efficiently used for embedded systems. In a future work, we will implement the entire Haar classifier in our architecture.

Acknowledgment

This research was supported by Multi-Ministry Collaborative R&D program (R&D program for complex cognitive technology) through the National Research Foundation of Korea (NRF) funded by the Ministry of Trade, Industry and Energy (NRF–2018M3E3A1057248).

References

[1] P. Viola and M. J. Jones, “Robust real-time face detection,” International journal of computer vision, vol. 57, no. 2, 2004, pp. 137–154.

[2] C. Kumar and S. Agarwal, “A novel architecture for dynamic integral image generation for Haar-based face detection on FPGA,” in TENCON 2014–2014 IEEE Region 10 Conference, 2014, pp. 1–6.

[3] D. Kim, J. Hyun and B. Moon, “Memory-Efficient Architecture for Contrast Enhancement and Integral Image Computation,” in 2020 IEEE International Conference on Electronics, Information, and Communication, 2020, pp. 1–4.

[4] H. J. Belt, “Word length reduction for the integral image,” in 2008 15th IEEE International Conference on Image Processing, 2008, pp. 805–808.