Implementation of Real-time Connected Component Labeling using FPGA

Tsung-Han Tsai, Yuan-Chen Ho and Chi-En Tsai
Dept. of Electrical Engineering, National Central University, Taiwan

***Abstract*--Connected Component Labeling (CCL) is one of the important process in the field of image processing. It can detect connected component in binary image and label them. This paper proposes a real-time single-scan CCL architecture and implementation in field-programmable gate array (FPGA) platform. This implementation has been completed on Xilinx Vertex-5 FPGA device and just used with the internal memory for storing component size and position rather than saving a whole image. It has been working at 60Hz for video of 640x480. The architecture runs in real-time while having reasonably low resource utilization, making integration with other real-time algorithms feasible.**

Keyword—CCL, FPGA, labeling, real-time processing

I. INTRODUCTION

CCL is used to detect connected objects in binary image, and this technique is commonly used in image recognition and tracking. The classic connected components algorithm [1-2] requires two-scan passes for the image. The first scan assigns the temporary labels to current pixels and records label equivalences. In the second scan, all equivalent labels are replaced through the representative label. These two-scan labeling algorithm are time consuming. In order to avoid this unnecessary repetition, [3-4] use one-scan algorithm, as shown in Figure 1. For the current pixel, four neighbors i.e. ABCD are examined which have been already processed. Once an unlabeled pixel is found, the current pixel is labeled immediately without the need of performing second scan. If any one pixel among ABCD is labeled, then the current pixel will be labelled with the smallest value among ABCD. Accordingly it will update the value in ABCD as the same label as new pixel's label. This update is performed after the scanning has been complete. In this paper, we extend the architecture while using internal memory to store component size and position which has not been done in [4].

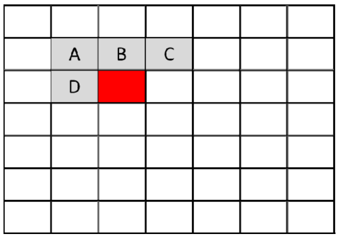

Figure 1. A label is assigned to the current pixel based on already processed neighbors.

II. HARDWARE ARCHITECTURE

Figure 2 illustrates a scenario where the CCL is utilized for object detection and window plotting in real-time. Image recognition or object tracking can be easily integrated to system to increase system functionality.

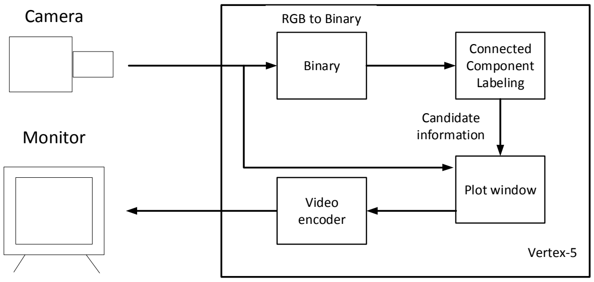

Figure 2. System architecture of CCL

Figure 3 shows the CCL architecture which has been modified from [4], where A, B, C and D blocks in Fig. 3 are the registers to save the neighbor's label (corresponding to Fig. 1). 'Label selection' block is a combinational logic block for label comparison. 'Row buffer' block is first-in-first-out (FIFO) to save the current line pixel's label after labeling. 'Merger table' block is a dual-port block RAM (BRAM) to replace the labeling and merger the cached pixel. 'Control unit' block is the finite-state machine (FSM) that controls image synchronization, image width, image height and *merger* signal. When *merger* signal becomes high, *combine* signal will become high and it will send signal to 'Component Information Operation' block to combine component information. 'Component Information Operation' block calculates the connected component's information where X/Y counter provides the position information. 'Data table' block is also implemented using a dual-port BRAM to save the component information. When *data_valid* becomes enable, 'Plot Window' module will read connected components position from Data table block for plotting windows on the candidate component.

III. EXPERIMENT RESULT

CCL processing time is calculated as following:

$$Number\ of\ clock\ cycles = (M+B)*N + 256-B$$

where M is image width, N is image height, B is line blank length and value of 256 presents maximum labeling number. When the input frame's resolution is 640 x 480 (clock @40MHz), and line blank is 256 cycles, the CCL processing time is 10.75ms per frame.

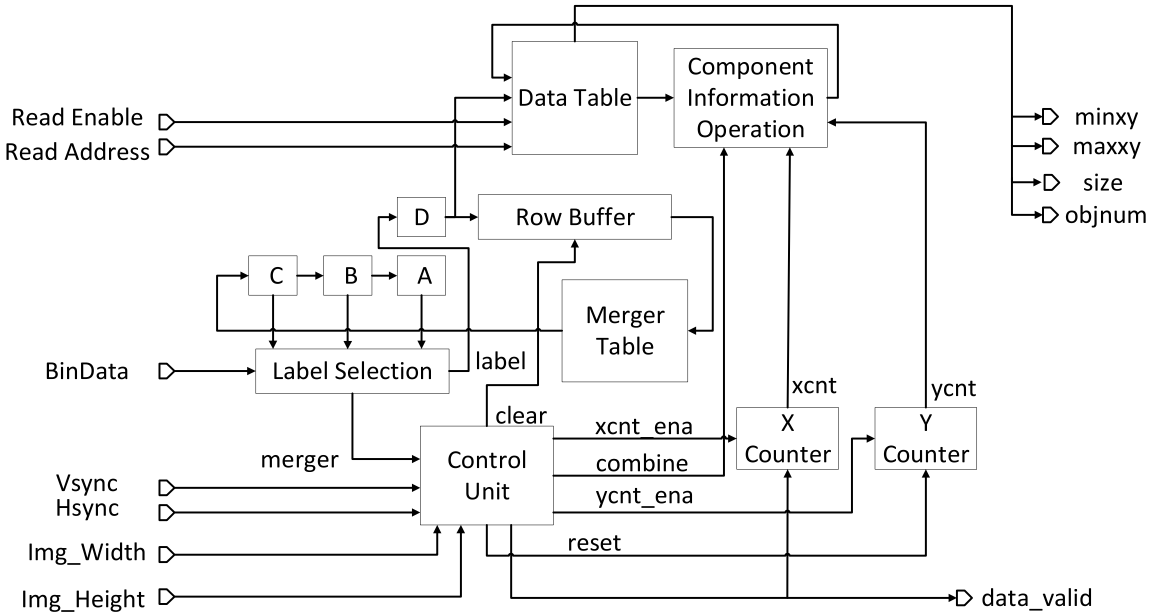

Figure 3. The architecture of the single pass algorithm.

The resource requirement and performance with other paper implementations is listed in TABLE I. This work provides higher operating frequency as compared to other CCL designs. [6] used the same FPGA platform as this study, but their resource utilization is higher than this work. [5] used less BRAM but just achieved a low working frequency.

The proposed design runs at speed of 128.07MHz as reported by Xilinx place and route (PAR) timing analysis. At this speed, the implementation is able to operate at high-resolution images in real-time. At 128.07 MHz, pixel cycle time is 7.8ns which makes it possible to label 223 frames per second for 640x480 image size, and label 94 frames per second for 1280x720 image size.

TABLE I. Comparison of other design.

	[5]	**[6]**	**This work**
Tech	Vertex 2	Vertex 5	Vertex 5
Image size	640x480	640x480	640x480
LUTs	1757	3136	1350
Register	600	3584	619
BRAM (bit)	72k	2131k	864k
FPS (Hz)	60	60	60
Fmax (MHz)	40.63	125	128

I. Conclusion

A highly sequential CCL algorithm has been design and implemented. This paper shows that proposed architecture with good performance as compared to other designs. This design has been implemented on FPGA without saving a whole image as the input. Since this implementation is work in real-time, it makes an integration feasible and easily to combine in some vision-based algorithms.

Acknowledgment

The authors would like to thank the editor and anonymous reviewers for their constructive comments which helped to improve the quality of this paper.

Reference

[1] A. Rosenfeld and J. Pfaltz, "Sequential Operations in Digital Picture Processing", *Journal of the ACM*, 13:(4) 471-494 ,1966.

[2] Kesheng Wu, Ekow Otoo, Kenji Suzuki, “Optimizing Two-Pass Connected-Component Labeling Algorithms”, *Pattern Anal*. Appl. 2008.

[3] AbuBaker. A, Qahwaji. R, Ipson. S, & Saleh. M, “One scan connected component labeling technique”, *IEEE International Conference on Signal Processing and Communications (ICSPC 2007)*, pp. 1283-1286, 2007.11.

[4] D. G. Bailey, C. T. Johnston, “Single Pass Connected Components Analysis”, *Proc. of Image and Vision Computing*, New Zeland, 2007.

[5] D. G. Bailey, C. T. Johnston, “Optimised Single Pass Connected Components Analysis”, *IEEE International Conference on Electrical & Computer Engineering (ICECE Technology 2008),*pp.185-192, 2008.12.

[6] J. G. Pandey, A. Karmakar, A. K. Mishra, C. Shekhar, “Implementation of an Improved Connected Component Labeling Algorithm using FPGA-based Platform”, *IEEE International Conference on Signal Processing and Communications (SPCOM 2014),* 2014.6.