Implementation of Binary Image Processing with Morphology Operation

Mageshwar. S[1], Saranya. P[2]
[1]PG Scholar, Sriguru Institute of Technology, Coimbatore-641 110, India
[2]Assistant Professor, ECE, Sriguru Institute of Technology, Coimbatore-641 110, India

Abstract- **Binary image processing is a powerful tool in many image and video processing applications, target tracking, multimedia application, and computer vision. Binary image processing chips have been designed to generalize the binary image applications of a chip. Chips were presented to progress basic binary morphological operations such as dilation, erosion, opening and closing. The processor's architecture is consists of a combination of reconfigurable binary processing module, input and output image control units. The reconfigurable binary processing module, which consists of fine and mixed-grained reconfigurable binary compute units and output control logic, works binary image processing operation especially mathematical morphology operations and implements related motion detection algorithms more than 237 frames per second for any image. The simulation and experimental results is suitable for real-time binary image processing applications.**

Keywords—Binary Image processor, field-programmable gate array (FPGA), mathematical morphology operation, mixed-grained, median filter.

1. INTRODUCTION

Binary image processing is a powerful tools and extremely used in different areas, such as object recognition, tracking, motion detection and machine intelligence [1]-[7], image analysis [8], video processing [10], computer vision, and identification & authentication systems [13]-[16]. Binary image processing has been commonly implemented using processors such as CPU or DSP.

High-speed implementation of binary image processing operations can be efficiently realized by using specialized chips for binary image processing. Therefore, binary image processing module chips have attracted much more attention in the field of image processing. The major drawback of application-specific chips is the lack of flexibility. The reconfigurable processing technique can bridge the gap between application-specific integrated circuits and flexibility.

A reconfigurable binary image processing system with high flexibility, performance, small size, and low power consumption can be implement in a single chip. Most reconfigurable vision chips can realize a reconfigurable computing by processing an element array [11], [12]. A reconfigurable image processing accelerator incorporating eight macro processing elements was designed to support real-time change detection and background registration based on video and object segmentation algorithm. Nowadays, a vision chip with the architecture of a massively parallel cellular array of processing elements was presented for image processing by using the asynchronous or synchronous processing technique [12]. It has been a common practice to build application-specific chips for real-time binary image processing [20]. Other general-purpose chips have the architecture of a digital processor array in which each digital processor handles one pixel.

When large sized images are processed, the chips will become extremely large. Further, these are needed to design a high performance, small size, and large application range chip for real-time binary image processing .This paper presents a binary image processor that consists of a reconfigurable binary processing module, including reconfigurable binary compute units and output control logic, input and output image control unit circuits. The processor is implemented to perform real time binary image processing. It is found that the processor can process pixel-level images and extract image features, such as boundary and motion detection of images. The processor has the advantages of high speed, simple structure, and various application ranges.

2. RELATED WORKS

2.1 Image analysis

It's only a slight oversimplification to say that the fundamental problem of image analysis is pattern recognition the purpose of which is to recognize image patterns corresponding to physical objects in the scene and determine their pose (position, orientation, size, etc.) [9]. Often the results of pattern recognition are all that's needed, for example a robot guidance system supplies an object's pose to a robot, and in other cases a pattern recognition step is needed to find an object so that it can be inspected for defects or correct assembly.

2.2 Pixel Mapping

Point transforms include a large set of enhancements that are useful with scalar-valued pixels (e.g. monochrome images). Often these are implemented by a single software routine (or hardware module) that uses a lookup table. Lookup tables are fast and can be programmed for any function offering the ultimate in generality at reasonable speed. Pixel maps are most useful when the function is computed based on global statistics of the image. One can process an image to have a desired gain and offset, for example, based on the mean and standard deviation, or alternatively, the minimum and maximum, of the input.

2.3 Noise Reduction Filtering

Noise reduction is the process of removing noise from a signal. All recording devices, both analog or digital have traits which make them susceptible to noise. Noise can be random or white noise with no coherence, or coherent noise introduced by the device's mechanism or processing algorithms.

In the case of photographic film and magnetic tape, noise (both visible and audible) is introduced due to the grain structure of the medium. In photographic film, the size of the grains in the film determines the film's sensitivity, more sensitive film having larger sized grains. In magnetic tape, the larger the grains of the magnetic particles (usually ferric oxide or magnetite), the more prone the medium is to noise.

3. MORPHOLOGY OPERATIONS

3.1 Dilation

Gradually enlarges the boundaries of regions of foreground pixels (i.e. white pixels, typically). Areas of foreground pixels grow in size while holes within those regions become smaller. Input to dilation operator: Image and structuring element [1], [3].

In other words, for each background (input) pixel, superimpose the structuring element with the input image. If at least one pixel in the structuring element coincides with a foreground pixel in the image underneath, then the input pixel is set to the foreground value. If all the corresponding pixels in the image are background however, the input pixel is left at the background value.

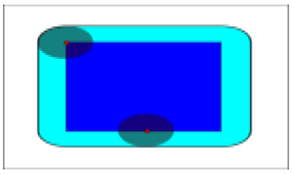

Figure1. Dilation

The dilation of A by the structuring element B is defined by:

$$A \oplus B = \bigcup_{b \in B} A_b$$

The dilation is commutative, also given by:

$$A \oplus B = B \oplus A = \bigcup_{a \in A} B_a$$

If B has a center on the origin, as before, then the dilation of A by B can be understood as the locus of the points covered by B when the center of B moves inside A. In the above example, the dilation of the square of side 10 by the disk of radius 2 is a square of side 14, with rounded corners, centered at the origin. The radius of the rounded corners is 2.

The dilation can also be obtained by:

$$A \oplus B = \{z \in E \mid (B^s)_z \cap A \neq \varnothing\}$$

Where B^s denotes the symmetric of B, that is,

$$B^s = \{x \in E \mid -x \in B\}$$

Example application: Dilation is the dual operation of the erosion. Figures that are very lightly drawn get thick when "dilated". Easiest way to describe it is to imagine the same fax/text is written with a thicker pen.

3.2 Erosion

The basic effect of the operator on a binary image is to erode away the boundaries of regions of foreground pixels (i.e. white pixels, typically). Shrink areas of foreground pixels in size and holes within those areas become larger.

In other words, For each foreground (input) pixel, superimpose the structuring element with the input image. The input pixel is left as it is if it is the foreground pixel in the structuring element. If any of the corresponding pixels in the image are background however, the input pixel is also set to background value.

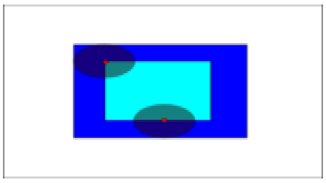

Figure2. Erosion

The erosion of the binary image A by the structuring element B is defined by:

$$A \ominus B = \{z \in E \mid B_z \subseteq A\},$$

Where B_z is the translation of B by the vector z,

$$B_z = \{b + z \mid b \in B\} \; \forall z \in E,.$$

When the structuring element B has a center (e.g., B is a disk or a square), and this center is located on the origin of E, then the erosion of A by B can be understood as the locus of points reached by the center of B when B moves inside A. For example, the erosion of a square of side 10, centered at the origin, by a disc of radius 2, also centered at the origin, is a square of side 6 centered at the origin. The erosion of A by B is also given by the expression:

$$A \ominus B = \bigcap_{b \in B} A_{-b}$$

Example application: Assume we have received a fax of a dark photocopy. Everything looks like it was written with a pen that is bleeding. Erosion process will allow thicker lines to get skinny and detect the hole inside the letter "o".

3.3 Opening

An Erosion followed by a dilation using the same structuring element for both operations.

Inputs: Image for opening and structuring element somewhat like erosion -it tends to remove some of the foreground (bright) pixels from the edges of object region and used to preserve foreground regions that have a similar shape to the structuring element.

Grey level opening consists simply of grey-level erosion followed by grey-level dilation. The opening of A by B is obtained by the erosion of A by B, followed by dilation of the resulting image by B:

$$A \circ B = (A \ominus B) \oplus B$$

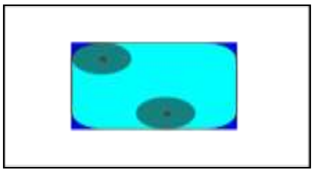

Figure 3 Opening.

The opening is also given by

$$A \circ B = \bigcup_{B_x \subseteq A} B_x$$

In which means that it is the locus of translations of the structuring element B inside the image A. In the case of the square of side 10, and a disc of radius 2 as the structuring element, the opening is a square of side 10 with rounded corners, where the corner radius is 2.

Example application: Let's assume someone has written a note on a non-soaking paper and that the writing looks as if it is growing tiny hairy roots all over. Opening essentially removes the outer tiny "hairline" leaks and restores the text. The side effect is that it rounds off things. The sharp edges start to disappear.

3.4 Closing

It's a dilation followed by erosion using the same structuring element for both operations. Enlarges the boundaries of foreground (bright) regions in an image Shrinks background color holes and Less destructive of the original boundary shape.

Operation determined by a structuring element. Preserves background regions that have a similar shape to the structuring element

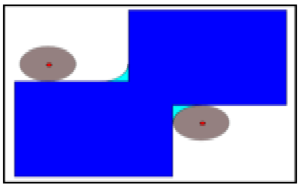

Figure 4 Closing.

The closing of A by B is obtained by the dilation of A by B, followed by erosion of the resulting structure by B:

$$A \bullet B = (A \oplus B) \ominus B.$$

The closing can also be obtained by

$$A \bullet B = (A^c \circ B^s)^c$$

In where X^c denotes the complement of X relative to E (that is,

$$X^c = \{x \in E | x \notin X\}).$$

The above means that the closing is the complement of the locus of translations of the symmetric of the structuring element outside the image A.

4. DESIGN AND IMPLEMENTATION

4.1 Design Criteria

Reconfigurable binary image processing chips have been designed to generalize the binary image applications of a chip. Chips were presented to perform basic binary morphological operations, such as dilation, erosion, opening, and closing. Programmable analog vision processors based on the cellular neural or nonlinear network universal machine architecture were proposed for a wide range of applications such as motion analysis and texture classification. A programmable single instruction multiple data (SIMD) real time vision chip was presented to achieve high-speed target tracking.

4.2 Block Diagram

A reconfigurable image processing accelerator incorporating eight macro-processing elements was designed to support real-time change detection and background registration based on video object segmentation algorithm. Recently, a vision chip with the architecture of a massively parallel cellular array of processing elements was presented for image processing by using the asynchronous or synchronous processing technique.

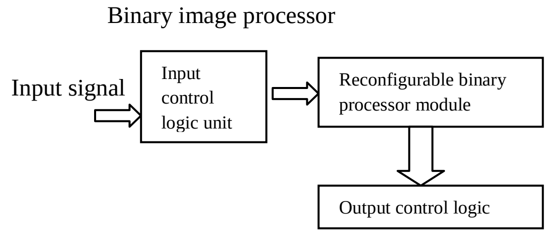

Figure5 Block diagram of binary image processor

The presented processor is designed for applications in image or video processing, computer vision, machine intelligence, and identification and authentication systems. Such systems should have a high flexibility and high performance processor for wide applications; therefore, the processor design is focused on high flexibility and speed. Some of the conventional works are designed for specific applications and some have large areas and high power consumption. Then, a reconfigurable binary processing module with high speed and simple structure is implemented for wide use and consuming fewer hardware resources.

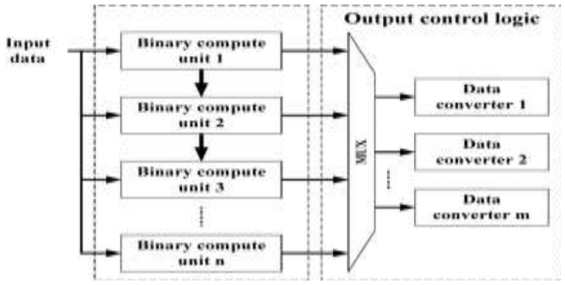

Figure6 Block Diagram of the reconfigurable binary processing module.

It can be divided into two main parts. The first part is the output control logic, which selects the output from all the binary compute unit outputs according to the given parameters and converts the series data of 1-b binary images into parallel data. The second part consists of several binary compute units that perform binary logic and binary image operations at a high speed. The binary image algorithms are realized by the operations in the individual binary compute units and the connection pattern of these units. The units can execute binary image operations in a pipelined or parallel manner. The operation executed in a binary compute unit is decided by configurable registers, including logic operation parameters, image resolution parameters; mask sizes, input and output selection parameters, and auxiliary parameters.

4.3 Binary Compute Unit

The architecture of the binary compute unit which has two binary compute elements and one set of operation elements can perform logic, reduction, median filtering and set operations. The binary compute unit has a mixed-grained architecture that has high flexibility,

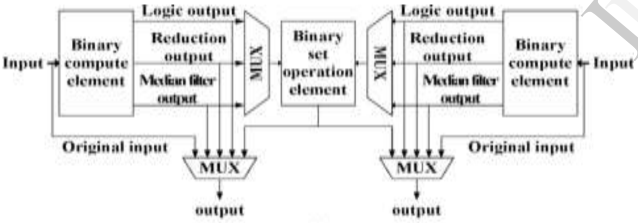

Figure7. Block Diagram of Binary Compute Unit.

Efficiency and performance. Granularity refers to the level of data manipulation. Usually, two types of granularity are distinguished: fine-grained, which corresponds to the bit-level manipulation of data and coarse-grained, which corresponds to the word level. The fine-grained architecture is highly flexible and the coarse-grained architecture has fewer reconfiguration parameters and is highly efficient.

The set operation element can perform binary set operations, such as union, intersection, complement, subtraction, addition, and straight-through output. The inputs of the set operation element and the outputs of the binary compute unit are transmitted via two sets of multiplexers, respectively, which makes the unit architecture more flexible. The inputs transmitted to the set operation element via the multiplexers can be the operation results of the binary logic elements, the reduction result, and the median filtering result. The outputs of the binary compute unit transmitted via multiplexers can be the original input of the binary compute unit, the operation results of the binary logic elements, the reduction result, the median filtering result, and the operation result of the set operation element. The set element has a fine-grained architecture. The operands of the set element are 1 b; therefore, the set element has a 1-b logic block and shows high flexibility and efficiency.

4.4 Binary Compute Element

The binary compute element comprises two input control multiplexers, n binary logic elements, a binary reduction element, and a binary median filter. The input control multiplexer selects input data for the binary logic element from the line memories, the SDRAM, and the parameters in the register group. When a video image is processed, line memories are needed to buffer image signals before they are input to binary logic elements. When the block size of the image to be processed is n × n, n − 1 line memories with a depth equal to the image width are needed to buffer the image signals. When images other than videos are processed, the input data are selected from the parameters in the register group or SDRAM. The binary logic element can perform operations such as AND, OR, NOT, NAND, NOR, XOR, XNOR, and straight-through output.

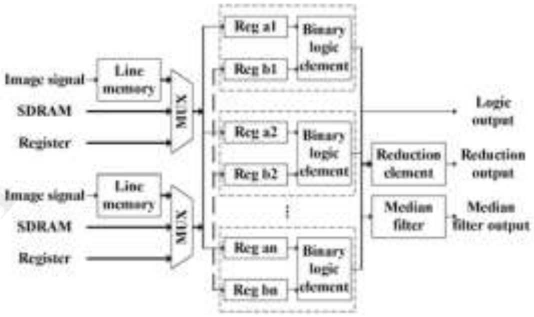

Figure8. Block Diagram of Binary Compute Element.

The reduction element performs operations such as reduction AND, reduction OR, reduction NAND, reduction NOR, reduction XOR, reduction XNOR, and straight through output. The set element performs operations such as union, intersection,

complement, subtraction, and XOR. All the operation results from the binary logic elements, the reduction element, and the binary median filter are synchronized and output via multiplexers to the next binary compute unit. The binary compute element has a coarse-grained architecture featured by high performance and short reconfigurable time. The binary compute unit has the characteristic of programmability and configurability since the programmable logic is applied in the design of the binary logic element, reduction element and binary median filter in the binary compute element, the set element, and the multiplexers. In sum, the binary compute unit is appropriate for binary image processing due to its high performance, flexibility, and short configuration time.

5. RESULTS AND DISCUSSIONS

As per the reconfigurable binary image processing architecure,the image is initallypassed through the input control logic unit. It can performthe some binary set of operation such as union, intersection, complement, substract, addition and straight through output. Where the blurring effects, salt and pepper noise are removed afterthat rank order filter is performed on the filtered output to obtain a scaled image.

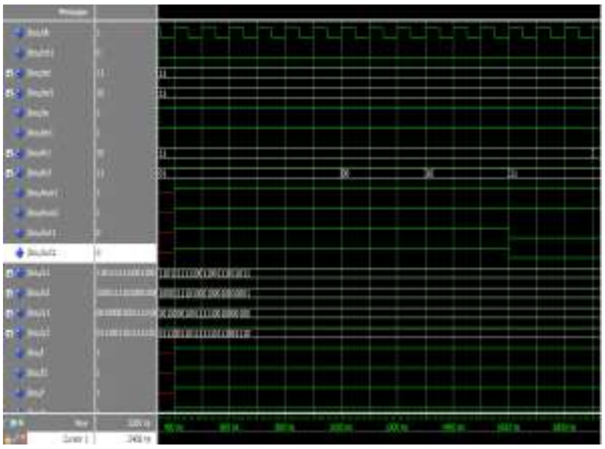

Figure9 Results for binary compute unit.

Above the figure shows result of binary compute unit. Binary compute unit performs binary set of operation element and binary compute element. Input image passed through the binary compute element and to perform the binary set of operation such as union, intersection, complement and addition. Binary compute element provides the output of median filter, reduction filter and logic outputs are passed through the binary compute unit via MUX. The binary compute unit can used to reduce size of the image and noise.

6. CONCLUSION

Hence a novel reconfigurable binary image processor technique is presented to develop a low-cost, low-power, low memory requirement, high flexibility and high performance for real-time applications. A vision system with high flexibility and performance, small size and low power consumption can be implemented in a single chip. When compared with the digital part, the analog part shows low robustness, accuracy and scalability although it has a small area and low power consumption. The dynamic reconfiguration approach was used to increase the processor performance. Basic mathematical morphology operations and complicated algorithms can easily be implemented on it because of its simple structure. Most reconfigurable vision chips realize reconfigurable computing by processing an element array. They can be used in image or video processing, target tracking, multimedia applications, and computer vision

REFERENCES

1. Bin Zhang, Kuizhi Mei and Nanning Zheng (2013), “Reconfigurable Processor for BinaryImageProcessing”, circuits and systems for video technology, vol. 23, no. 5
2. Chan. M, Chang. J, Chen. T (2009), “Efficient content analysis engine for visual surveillance network,” vol. 19, no. 5, pp. 693–703
3. Chien. S and Chen. L (2011), “Reconfigurable morphological image processing accelerator for video object segmentation,” vol. 62, no. 1, pp. 77–96
4. Dominguez-Castro. R, Espejo.S, Rodriguez-Vazquez. A (1997), “A 0.8-μm CMOS 2-D programmable mixed-signal focal-plane array processor with on-chip binary imaging and instructions storage,” vol. 32, no. 7, pp. 1013–1026
5. Dailey. D. J, Cathey F. W and Pumrin. S (2000), “An algorithm to estimate mean traffic speed using uncalibrated cameras,” vol. 1, no. 2, pp. 98–107
6. Fujii. K, Nakanishi. M, Shigematsu. S(2002), “A 500-dpi cellular-logic processing array for fingerprint-image enhancement and verification,” pp. 261–264
7. Ikenaga. T and Ogura. T (2000), “A fully parallel 1-Mb CAM LSI for real-time pixel-parallel image processing,” vol. 35, no. 4, pp. 536–544
8. Kim. J, Park. J, Lee. K (2007), “A portable surveillance camera architecture using one-bit motion detection,” vol. 53, no. 4, pp. 1254–1259
9. Laiho. M, Poikonen. J and Paasio. A (2009), Space-dependent binary image processing within a 64×64 mixed-mode array processor
10. Lipton. A. J, Fujiyoshi. H and Patil. R. S (1998), “Moving target classification and tracking from real-time video,” pp. 8–14
11. Liu.Y and Pomalaza-Raez. C (2009), “A low-complexity algorithm for the on-chip moment computation of binary images,” pp. 1871–1876
12. Lopich. A and Dudek. P (2011), “A SIMD cellular processor array vision chip with asynchronous processing capabilities,” vol. 58, no. 10, pp. 2420–2431
13. Lyu. M. R, Song. J and Cai. M (2005), “A comprehensive method for multilingual video text detection, localization, and extraction,” vol. 15, no. 2, pp. 243–255
14. Miao. W, Lin. Q, Zhang. W (2008) et al., “A programmable SIMD vision chip for real-time vision applications,” vol. 43, no. 6, pp. 1470–1479
15. Malamas. E. N, A. G. Malamos. A. G and T. A. Varvarigou. T. A (2000), “Fast implementation of binary morphological operations on hardware-efficient systolic architectures,” vol. 25, no. 1, pp. 79–93
16. Park. H. J, Kim. K. B, Kim. J. H (2007) et al., “A novel motion detection pointing device using a binary CMOS image sensor,” pp. 837–840
17. Miao. W, Lin. Q, Zhang. W (2008) et al., “A programmable SIMD vision chip for real-time vision applications,” vol. 43, no. 6, pp. 1470–1479
18. Pedrino. E. C, Moran din. O and Roda. V. O (2011), “Intelligent FPGA based system for shape recognition,” pp. 197–202
19. Pedrino. E. C, Saito. J. H, and Roda. V. O (2010), “Architecture for binary mathematical morphology reconfigurable by genetic programming,” pp. 93–98
20. Shaaban K. M, Ali. S. A, and Mahdy. Y. B (1999), “A chip design for binary and binary morphological operations,” pp. 554–559
21. Talu M. F and Turkoglu. I (2009),“A novel object recognition method based on improved edge tracing for binary images,”pp.1–5