A CELLULAR-AUTOMATON-TYPE IMAGE EXTRACTION ALGORITHM AND ITS IMPLEMENTATION USING AN FPGA

T. Nakano[†] *, T. Morie*[‡] *, M. Nagata*[*] *, and A. Iwata*[†]

[†] Nakano and Iwata's Dept
Graduate School of Advanced Sciences of Matter, Hiroshima University
Higashi-Hiroshima, 739-8526 Japan
Phone:+81-824-24-7686, Fax:+81-824-22-7195,
Email:{teppei,iwa}@dsl.hiroshima-u.ac.jp

[‡] Morie's Dept
Graduate School of Life Science and Systems Engineering, Kyushu Institute of Technology
Hibikino, Wakamatsu-ku, Kitakyushu 808-0196 Japan
Phone:+81-93-695-6122, Fax:+81-93-695-6014,
Email: morie@brain.kyutech.ac.jp

[*] Nagata's Dept
Computer and Systems Engineering, Kobe University
Nada-ku, Kobe, 657-8501 Japan
Phone:+81-78-803-6569, Fax:+81-78-803-6221
Email:nagata@cs.kobe-u.ac.jp

ABSTRACT

This paper proposes a new region extraction algorithm based on cellular automaton operation, which only utilizes the region boundary information of the image. A simple pixel circuit for pixel-parallel operation is also proposed. Logic simulation results of using Verilog-HDL indicate that the proposed algorithm is about 100 times faster than the serial labeling processing for 100×100-pixel images. An experimental result for a 30×30-pixel image using an FPGA chip demonstrates that all regions are successfully extracted one by one within 6 μsec with a clock frequency of 25 MHz.

1. INTRODUCTION

In order to recognize natural scene images, which usually include several objects, we should segment the image into recognition target object regions and extract them one by one. In that case, coarse region segmentation that ignores small parts in each object region must be performed. We have already demonstrated that the resistive-fuse network can perform such coarse region segmentation [1]. However, the resistive-fuse network cannot extract each segmented region automatically. Thus, it requires a region extraction method.

This work was supported by the Ministry of Education, Science,Sports, and Culture under Grant-in-Aid for Scientific Research on Priority Areas (A).

In this paper, we propose a region extraction algorithm based on cellular-automaton-type pixel-parallel processing. More specifically, in the proposed method, each region surrounded by boundary pixels is sequentially extracted, which makes it possible to perform pipeline processing for the following image processing. In contrast, the usual segmentation processing performed in DSPs, which is called *labeling*, cannot achieve pixel-parallelism.

We have already proposed another method for the same purpose, which uses a resistive-fuse network and a nonlinear oscillator network [2]. The extraction method proposed here is much simpler and can be implemented even in digital circuits.

2. CELLULAR-AUTOMATON-TYPE REGION EXTRACTION ALGORITHM

We assume that each pixel has three states: *unfired*, *firing*, *fired*, and that the region boundary data is already obtained before the extraction processing. The extraction algorithm is schematically shown in Fig. 1, and numerical simulation results using MATLAB are shown in Fig. 2. The detail of the algorithm is given below with digital circuit implementation shown in the corresponding figures.

(1) Initialization: if a pixel is at boundaries, then set its state *fired*, otherwise set it *unfired* (Fig. 1(a):Fig. 3).

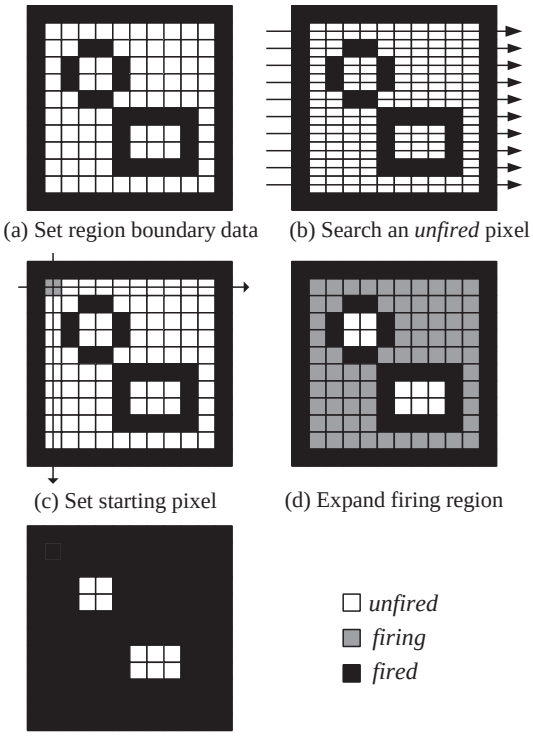


Figure 1: Region extraction algorithm.

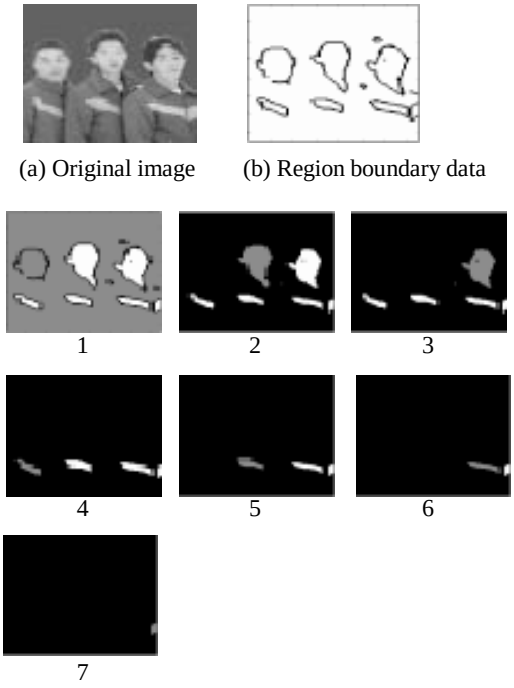


Figure 2: Numerical simulation results using MATLAB. The white, gray, and black regions are *unfired*, *firing*, and *fired*, respectively.

(2) Detect an *unfired* pixel, which we call a starting pixel, and set its state *firing* (Fig. 1(b)(c):Fig. 4).

(3) For every *unfired* pixel, if any neighboring pixel is *firing*, set the pixel state *firing* (Fig. 1(d):Fig. 5).

(4) Repeat process (3) until the total state change stops. For detecting the end of the firing process, we have to store the just previous state (*unfired* or *firing*) (Fig. 6).

(5) Read out the addresses of all the *firing* pixels or the boundary of the *firing* region (Fig. 7, Fig. 8).

(6) Set all *firing* pixels *fired* (Fig. 1(e)).

(7) Repeat process (2)-(6) until no *unfired* pixel is detected.

In the above processes, (1), (3), (6) can be performed in pixel-parallel. Processes (2), (4), (5), (7) can be performed in row- (or column-) parallel if detection lines are formed in order to detect the states of all pixels belonging to the same row simultaneously. This can easily be achieved by using current summing lines in the real circuit. Thus, if the image size is $M \times N$ pixels, this algorithm can extract all regions during a period on the order of $M+N$; on the other hand, for usual *labeling* processing requires $M \times N$ processing time.

The proposed algorithm can be considered as a simple version of the "region growing" method [3] that is used for gray-scale image segmentation. In our approach, however, boundary detection of gray-scale images is performed by resistive-fuse networks, and the region extraction algorithm proposed here is applied to images having only boundary information.

3. DIGITAL CIRCUIT IMPLEMENTATION

A pixel circuit implementing the above algorithm consists of a three-bit register, an exclusive-OR gate and four switches as shown in Fig. 9. This circuit can easily be incorporated into the pixel circuit of resistive-fuse networks for image segmentation proposed before [4].

Here, we demonstrate the performance of the algorithm by implementing it in an FPGA. The RTL (register-transfer level) simulation results indicate that the processing time for region extraction is proportional to the square root of the total number of pixels as shown in Fig. 10. From these results, processing time of less than 20 μs is needed in 25 MHz operation for images with 100×100 pixels. This is more than 100 times faster than serial labeling processing.

An experimental result using an FPGA chip (ALTERA /EP20K400EFC672-1X) with 25 MHz is shown in Fig. 11. We can successfully extract 5 regions in a 30×30-pixel image within 6μs. Obviously, real images have much more and complexer regions than the example used in this experiment, but preprocessing by a resistive-fuse network achieves coarse region segmentation and a limited number of regions are

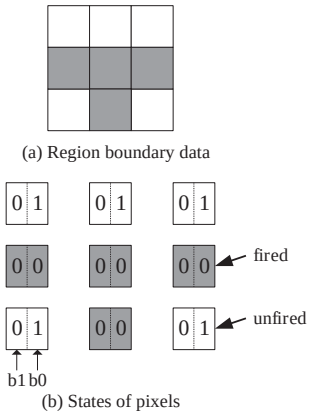


Figure 3: Initialization (3×3 pixel case).

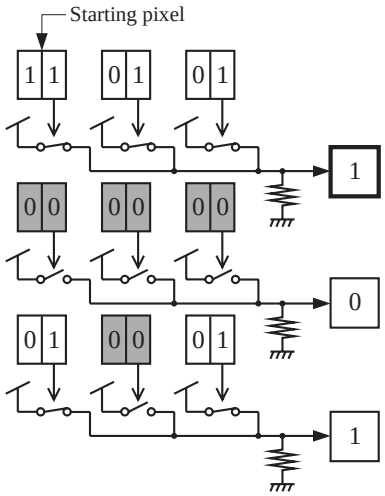


Figure 4: Determination of starting pixel.

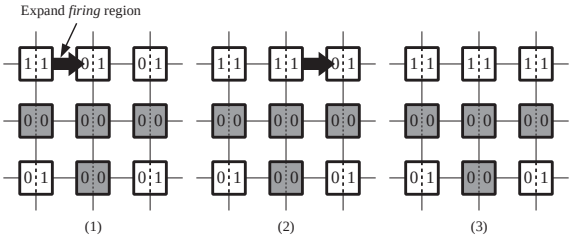


Figure 5: Expansion of firing region.

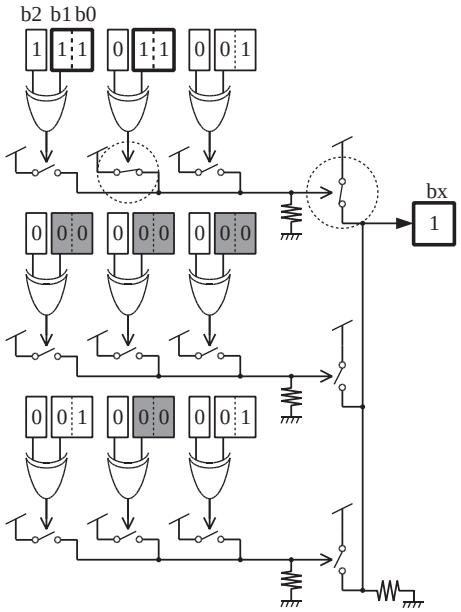


Figure 6: Detection of the end of firing process. Register "b2" holds the one-clock previous state. Output "bx" is "1" if firing region expansion continues.

segmented, as shown in Fig. 2. Therefore, this experimental result is valid for real image processing.

4. CONCLUSION

We proposed a region extraction method based on cellular automaton operation and a simple pixel circuit for pixel-parallel operation. FPGA implementation for image extraction of a 30×30-pixel image demonstrated that all regions were successfully extracted one by one.

This simple circuit can be incorporated into a resistive-fuse network circuit that we previously proposed for coarse region segmentation.

REFERENCES

[1] T. Morie, M. Nagata, and A. Iwata, "Design of a Pixel-Parallel Feature Extraction VLSI System for Biologically-Inspired Object Recognition Methods," in *Proc. Int. Symp. on Nonlinear Theory and its Applications (NOLTA2001)*, pp. 371–374, Oct. 2001.

[2] H. Ando, T. Morie, M. Miyake, M. Nagata, and A. Iwata, "Image Segmentation/Extraction Using Nonlinear Cellular Networks and their VLSI Implementation Using Pulse-Modulation Techniques," *IEICE Trans. Fundamentals.*, vol. E85-A, no. 2, pp. 381–388, 2002.

[3] R. Adams and L. Bischof, "Seeded Region Growing," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 16, no. 6, pp. 641–647, 1994.

[4] T. Morie, M. Miyake, M. Nagata, and A. Iwata, "A 1-D CMOS PWM Cellular Neural Network Circuit and Resistive-Fuse Network Operation," in *Ext. Abs. of Int. Conf. on Solid State Devices and Materials*, pp. 90–91, Sept. 2001.

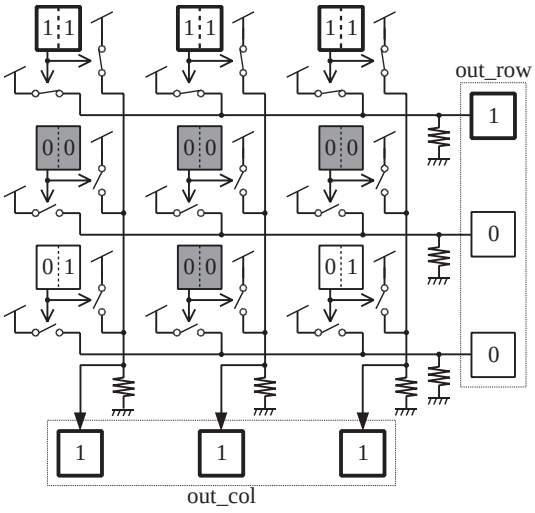
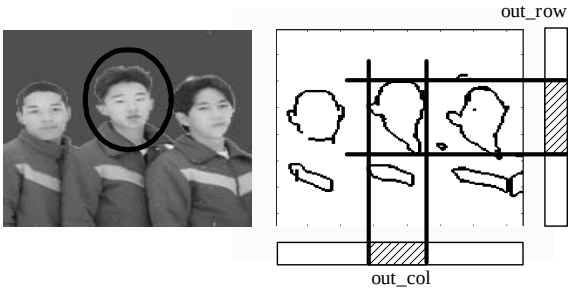


Figure 7: Readout of firing region



(a) Original image (b) Result

Figure 8: Example of real image for region readout.

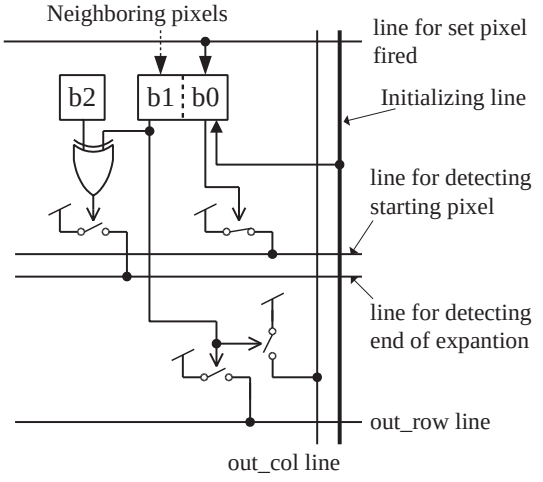


Figure 9: Pixel circuit for region extraction.

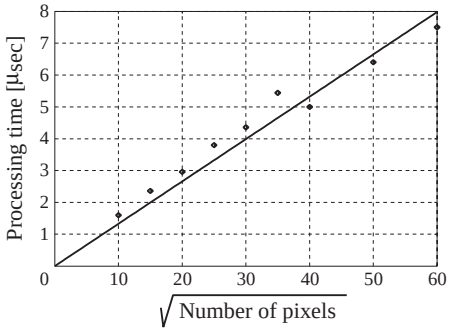


Figure 10: Relationship between image size (number of pixels) and processing time for 25 MHz operation (RTL simulation results).

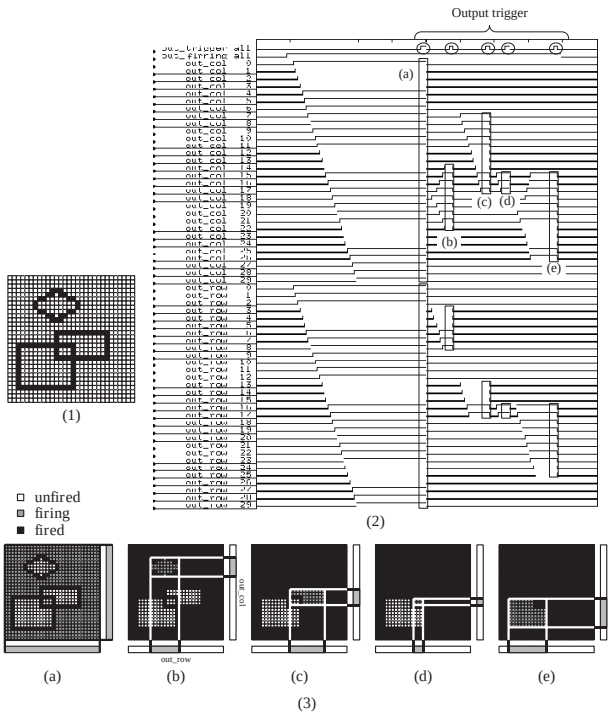


Figure 11: Experimental result for a 30×30-pixel image using an FPGA chip with 25 MHz. (1) Input image (boundary pixel data), (2) outputs obtained from logic analyzer, (3) schematic snapshots at region extraction timing.