Coarse Image Region Segmentation Using Resistive-fuse Networks Implemented in FPGA

Teppei NAKANO[†], Hiroshi ANDO[*], Hideaki ISHIZU[‡], Takashi MORIE[†], and Atsushi IWATA[*]

[†] Graduate School of Life Science and Systems Engineering,
Kyushu Institute of Technology,
2-4, Hibikino, Wakamatsu-ku, Kitakyushu, 808-0196 Japan
[*]Graduate School of Advanced Sciences of Matter, Hiroshima University,
1-3-1, Kagamiyama, Higashi-Hiroshima, 739-8526 Japan
[‡] Western Hiroshima Prefecture Industrial Research Institute
2-10-1, Aga-minami, Kure, Hiroshima, 737-0004 Japan

ABSTRACT

Two digital LSI implementation methods for nonlinear resistive networks are proposed; one is for pixel-parallel operation and the other is for pixel-serial operation. we have designed digital circuits that emulate the operation of analog nonlinear resistive networks by discrete-time dynamics based on clock operation. The steady state of the networks is obtained by repeating the updating sequence. The resistive-fuse network has been implemented in an FPGA. Coarse region segmentation of real images with 64×64 pixels at the video rate is successfully demonstrated by using an FPGA.

Keywords: Resistive-Fuse Network, Coarse Region Segmentation, FPGA Implementation

1. INTRODUCTION

The resistive-fuse network is a well-known image segmentation processing model in which image edges are preserved and noise is eliminated [1]. Some attempts for its analog LSI implementation have been proposed [2, 3]. Although the analog approach achieves compact circuit size and high processing speed, a practical design for large-scale resistive-fuse network circuits (more than 100 × 100 pixels) is very difficult to realize because of unexpected parasitic components and various disturbances such as noise, interference, and device parameter mismatch.

To overcome these difficulties, we have already proposed pulse-width-modulation cellular neural network (PWM-CNN) circuits, which can implement the resistive-fuse network model with high controllability. [4-6].

In this paper, in order to apply the resistive-fuse networks to digital image processing, we propose efficient digital implementation methods for nonlinear resistive networks. We demonstrate FPGA implementation of resistive-fuse networks for coarse image region segmentation.

2. COARSE REGION SEGMENTATION BY RESISTIVE-FUSE NETWORKS

Resistive-Fuse Networks with Discrete-Time Dynamics

Figure 1(a) shows a circuit diagram of the resistive-fuse network, in which nonlinear resistance elements connect the neighboring pixel nodes. The image input I_i for pixel i is given as a voltage source. The processing result (output) O_i is given as a node voltage. If the difference voltage between neighboring pixels $|O_j - O_i|$ is smaller than the threshold value δ, the network behaves as a simple linear resistive network, and image smoothing is performed. On the other hand, if $|O_j - O_i| \geq \delta$, the pixel nodes are disconnected each other, and such pixels are recognized as an image edge.

Moreover, by changing I-V characteristics of the nonlinear resistance element from linear resistance to resistive-fuse, as shown in Fig. 1(b), coarse region segmentation can be achieved. In this processing, a whole face region is segmented irrespective of small facial parts such as brows, eyes, a nose, and a mouth, as shown in Fig. 1(c).

In our approach proposed in this paper, the pure analog circuit shown in Fig. 1(a) is not used. Instead, we design a digital circuit that emulates the operation of the analog nonlinear circuit by discrete-time dynamics based on clock

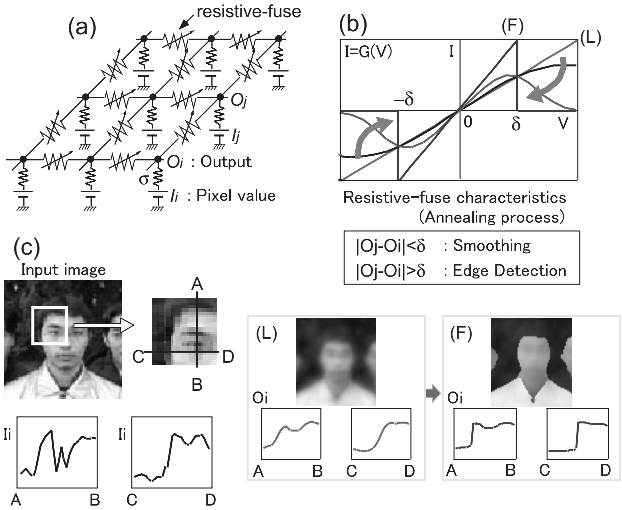

Figure 1: Resistive-fuse network circuit and the principle of coarse image region segmentation by resistive-fuse networks. (a) Resistive-fuse network circuit. (b) I-V characteristics of nonlinear resistance element, (L): linear resistance, (F): resistive-fuse. (c) Image processing results, (L) and (F) correspond to the characteristics in (b).

operation. The change in each node voltage is calculated using Kirchhoff's current law, and the steady state is obtained by repeating the updating process. The dynamics is expressed as follows:

$$O_i(t+1) - O_i(t) = \nu\Big[\sum_{j\in N_i} G(O_j - O_i) + \sigma(I_i - O_i)\Big] \quad (1)$$

where N_i represents neighboring pixels of i; ν is a constant; σ is a constant that expresses the conductance; $G(\cdot)$ is a voltage-current characteristic of the nonlinear resistance element, which is assumed to be an odd function, as shown in Fig. 1(b).

Robust Coarse Region Segmentation

By detecting the resistive-fuse elements that are *blown out* ($|O_j - O_i| \geq \delta$), we can obtain the edge information by coarse region segmentation. However, the contours of the obtained edge information often have discontinuity because of low contrast of the input image. We propose two methods for avoiding such discontinuity. One is to use color pixel data. For example, as shown in Fig. 2, if we combine the results from RGB data, more robust segmentation is obtained; the edge contour has less discontinuity. The other method is to use conventional image processing such as dilation (expansion), erosion, skeletonization. Figure 2 also shows a process to obtain continuous edge contours by using such image processing.

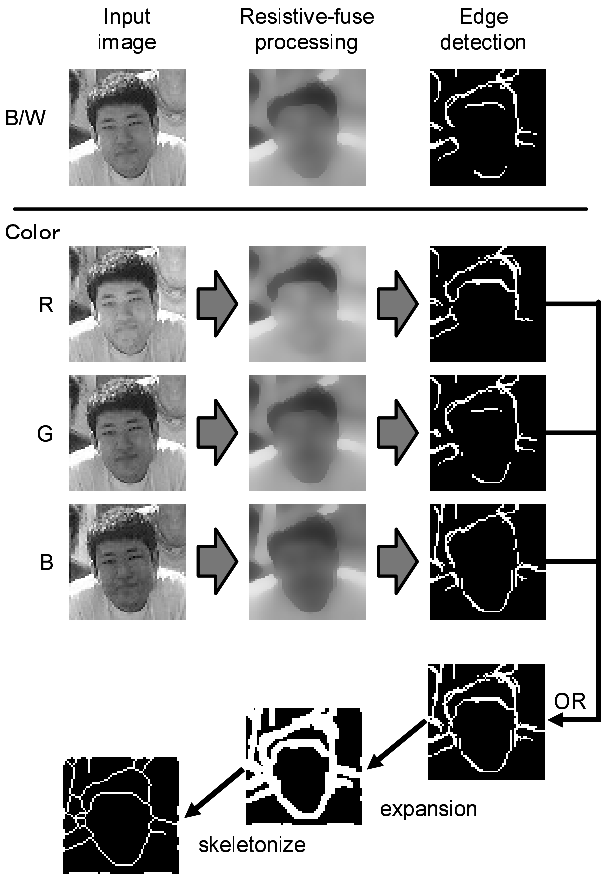

Figure 2: Coarse region segmentation using RGB images compared with the case of a black and white (B/W) image. A method to obtain complete closed edge contours using conventional image processing is also shown.

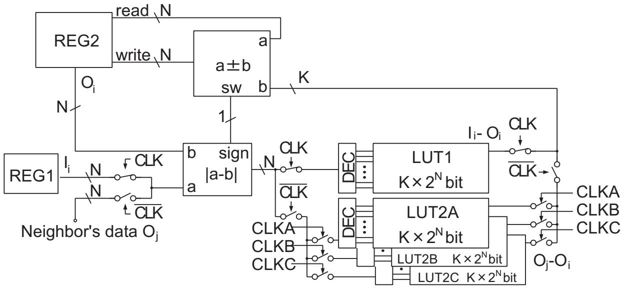

Figure 3: Digital pixel circuit implementing resistive-fuse networks

3. DIGITAL IMPLEMENTATION FOR RESISTIVE-FUSE NETWORKS

Pixel Circuit for Pixel-Parallel Implementation

One possible implementation method is based on the pixel-parallel approach. Figure 3 shows a pixel circuit for digital implementation of nonlinear networks such as the above-mentioned resistive-fuse networks with an annealing process. The input I_i and output O_i are stored at registers REG_1 and REG_2 with a bit precision of N, respectively. Here, N is set at 6-8. The data of I_i may be used as the initial value of O_i. The data stored in REG_2 (O_i) is updated according to the discrete-time dynamics given by Eq. (1). The lookup table memory LUT_1 transforms N-bit input data x to K-bit output data y according to linear function $y = \sigma x$. The bit precision K is determined by the transformation range, and it is typically 5-6. The lookup table memories LUT_{2A}–LUT_{2C} correspond to the nonlinear functions $G(\cdot)$ shown in Fig. 1(b).

The updating process proceeds as follows:

1) Set $CLKA = High$ and $CLKB = CLKC = Low$.
2) Set $CLK = High$ and transform the absolute value of the difference $|I_i - O_i|$ to $\sigma|I_i - O_i|$ by LUT_1, and update O_i in REG_2 using the transformed result and the sign of $(I_i - O_i)$.
3) Set $CLK = Low$ and transform $|O_j - O_i|$ to $G(|O_j - O_i|)$ by LUT_{2A}, and update O_i in REG_2 using the transformed result and the sign of $(O_j - O_i)$ based on the data of each neighboring pixel ($j \in N_i$).
4) Repeat steps 2 and 3, R times. (typically, $R = 30$)
5) Set $CLKA = CLKC = Low$, $CLKB = High$, and execute step 4.
6) Set $CLKA = CLKB = Low$, $CLKC = High$, and execute step 4.

The pixel-parallel implementation may lead to very high-speed processing, but the number of pixels included in a chip is limited. For real images with a large number of pixels, pixel-serial implementation is suitable.

Resistive-fuse Circuit for Pixel-Serial Implementation

In our pixel-serial implementation, each pixel data of the input image is serially processed by one processing circuit. Fig. 4 shows the memory assignment for sequential processing. Each pixel data of the input image is fed into the memory by raster scan. The pixel data, I_i, is stored in the source memory, and the processed result O_i is stored in the destination memory. It stores I_i as the initial data before resistive-fuse processing. Timing adjustment is achieved through registers and one-line memory, which is a FIFO memory that stores pixel data of one row of the image.

Figure 5 shows a circuit block diagram for pixel-serial implementation, in which data stored in memory units $a - i$ and $Smem$ are used in parallel. In the figure, the rectangular blocks with a clock input indicated by ">" operate synchronously with the system clock, and those without a clock input operate asynchronously. The synchronous blocks located at the same vertical position operate at the same clock timing in parallel. Because the data in $a - i$ and $Smem$ change every clock cycle, pipe-line processing is effectively performed. The number of clock cycles to obtain the updating result of the target pixel e is equal to that of register stages. The updating process is the same as that described in the previous section except that the differences between the data of target pixel e and those of the other pixels $a - d, f - i, Smem$ are simultaneously calculated.

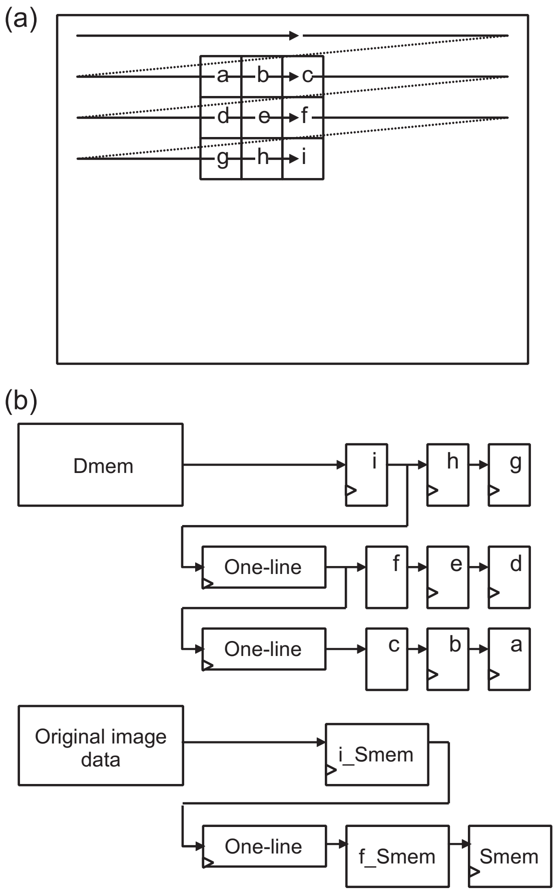

Figure 4: (a): Pixels assigned for processing (3×3 pixels) in the whole image pixels, and (b): memory units $a - i$ and source memory unit $Smem$ store the data of 3×3 processed pixels $a - i$ and the input pixel data, respectively.

Here, a key of this digital implementation is bit-shift operation. The M-bit shifted data of e is used for the calculation of updating values. This leads to effectively higher calculation precision in updating operation. In our

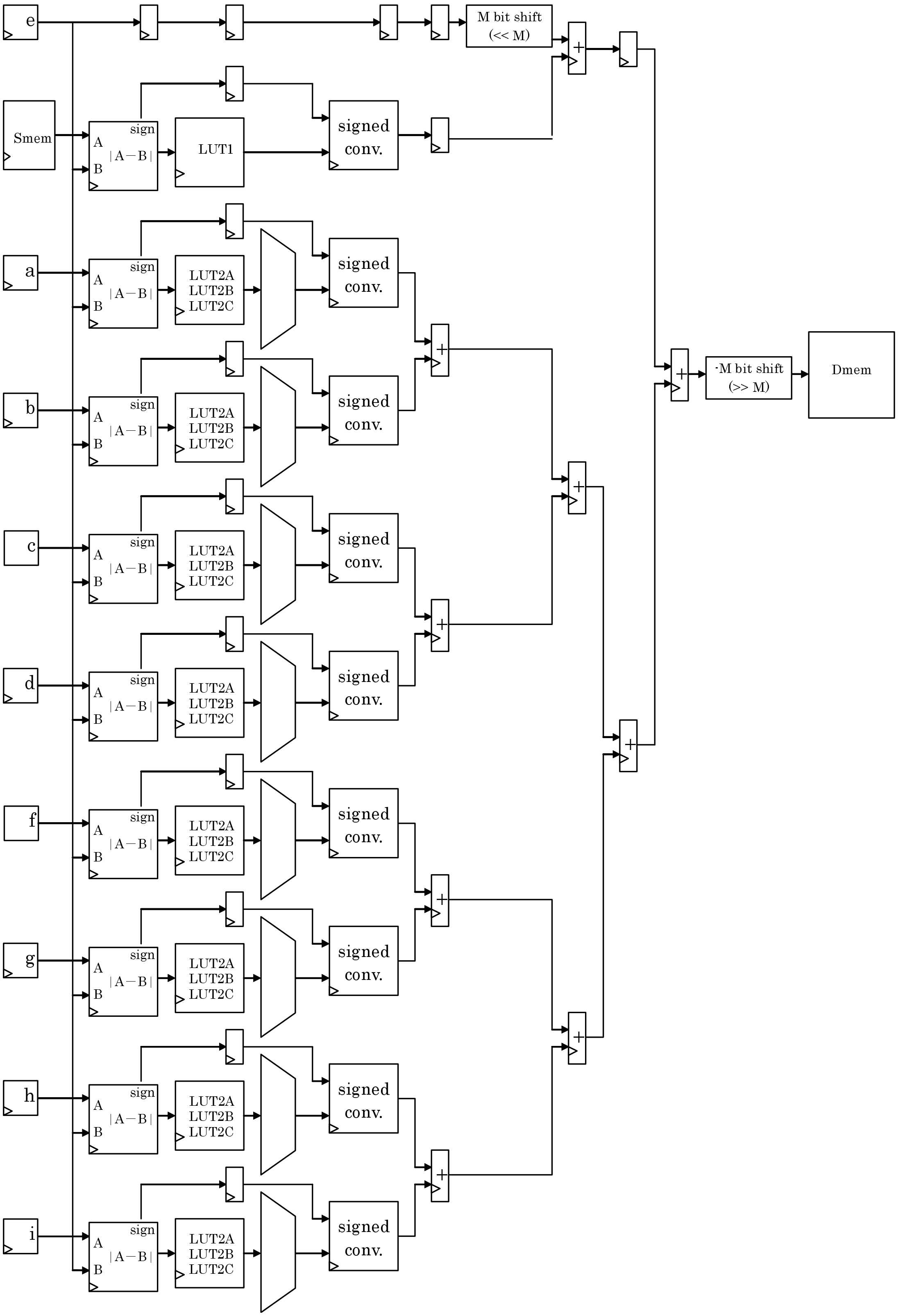

Figure 5 Resistive-fuse circuit for pixel-serial implementation

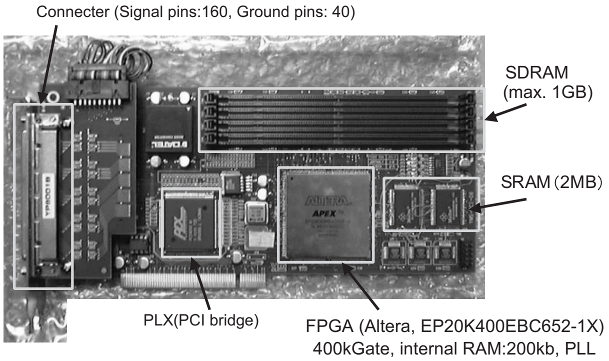
Connecter (Signal pins:160, Ground pins: 40)
SDRAM (max. 1GB)
SRAM（2MB）
PLX(PCI bridge)
FPGA (Altera, EP20K400EBC652-1X) 400kGate, internal RAM:200kb, PLL

Figure 6 PCI board including an FPGA.

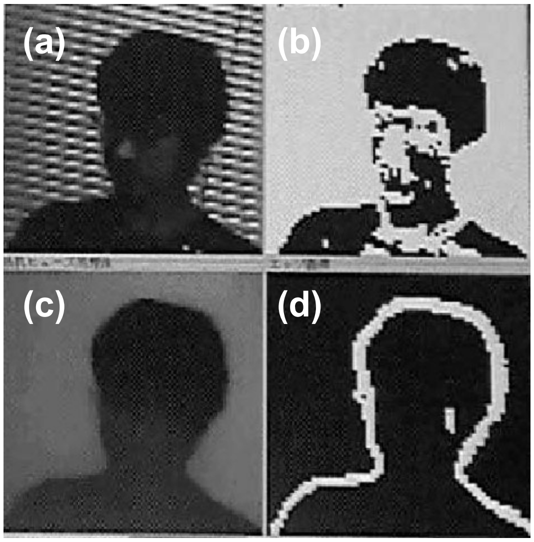

Figure 7: Demonstration of coarse image region segmentation by resistive-fuse processing implemented in an FPGA. (a) Input image, which is a snapshot of a person standing in front of a window-shade. (b) Usual edge detection result, where all edges of the detail of the face and the shade. No meaningful region can be segmented. (c) Resistive-fuse processing result. (d) Edge detection result of the resistive-fuse processing result, where only the whole human region is successfully segmented.

experiments, by setting $M = 3$, the image resolution N was able to be lowered by one bit, which can reduce memory size for storing image data.

4. FPGA IMPLEMENTATION AND COARSE REGION SEGMENTATION EXPERIMENT

The proposed algorithm was implemented using an FPGA. The FPGA used is ALTERA EP20K400 EBC652-1X, which is included in a PCI board shown in Fig. 6. Coarse region segmentation of real images was demonstrated as shown in Fig. 7. When the clock frequency was 40 MHz, the processing time was less than 20 ms for an image of 64×64 pixels.

5. CONCLUSIONS

Two digital LSI implementation methods for nonlinear resistive networks were presented, and the resistive-fuse network was implemented in an FPGA. Coarse region segmentation of real images with 64×64 pixels at the video rate was successfully demonstrated by using an FPGA.

ACKNOWLEDGMENT

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

REFERENCES

[1] J. Harris, C. Koch, and J. Luo, “Resistive fuses: Analog hardware for detecting discontinuities in early vision,” in **Analog VLSI Implementation of Neural Systems**, C. Mead and M. Ismail, Eds. Kluwer Academic Publishers, 1989, pp. 27–55.

[2] P. C. Yu, S. J. Decker, H. S. Lee, C. G. Sodini, and J. L. Wyatt, Jr., “CMOS resistive fuses for image smoothing and segmentation,” **IEEE J. Solid-State Circuits**, Vol. 27, No. 4, 1992, pp. 545–553.

[3] T. Sawaji, T. Sakai, H. Nagai, and T. Matsumoto, “A floating-gate MOS implementation of resistive fuse,” **Neural Computation**, Vol. 10, No. 2, 1998, pp. 485–498.

[4] T. Morie, M. Miyake, S. Nishijima, M. Nagata, and A. Iwata, “A multi-functional cellular neural network circuit using pulse modulation signals for image recognition,” in **Proc. Int. Conf. on Neural Information Processing (ICONIP)**, Taejon, Korea, Nov. 2000, pp. 613–617.

[5] 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 (SSDM)**, Tokyo, Sept. 2001, pp. 90–91.

[6] 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, 2002, pp. 381–388.