A THINNING PROCESS FOR AN IMPLEMENTATION IN A PIXEL ARRAY CIRCUIT

Kuo-Ting Wu and Chunyan Wang

Department of Electrical and Computer Engineering
Concordia University
1455 de Maisonneuve Blvd. West
Montreal, Quebec, Canada H3G 1M8
E-mail: {kt_wu, chunyan}@ece.concordia.ca

ABSTRACT

In this paper, a thinning process suitable for pixel array circuit implementation is presented. The process is designed to have the computations in parallel and to include only simple logic operations so that the structure of each pixel will be simple and the array circuit will have an acceptable resolution. Simulation with images of different patterns, including fingerprints, demonstrated that the proposed thinning process is effective, and its results meet the requirement of the minutia detection for identification. The structure of such a pixel array circuit is also designed as an example of implementation and presented in the paper. The circuit has four interconnections per pixel, and can be implemented in a digital CMOS process.

1. INTRODUCTION

Thinning process is often used in image processing systems. In particular, it is included in fingerprint image processing and a well-thinned ridge map can facilitate the minutiae detection for personal-identification[1]. Most thinning processes require extensive computations and are not developed for IC implementation. Hence, the objective of the work presented in this paper is to develop a thinning process implementable in a 2-D pixel array circuit, as shown in Fig. 1, to minimize the processing time, circuit area and power dissipation of a fingerprint process system.

A thinning process is applied to a binary fingerprint image. Many of existing methods for such an image thinning are based on differential computations on the adjacent pixels. An algorithm proposed in [2] is to compute the sum of the eight signals, each of which is the difference between two adjacent pixels in a 3×3 window, and compare the result with 2. Based on the same idea, the thinning algorithm reported in [3] provides more precisely defined conditions and it also simplifies the computation process. Some later algorithms based on [2] and [3] have been reported with emphasis in different aspects [4] [5].

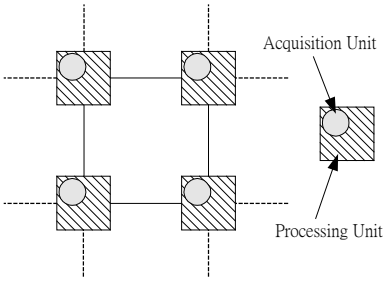

Fig. 1 Pixel array structure. The pixels are connected according to a well-defined detection method. The connections can be controlled by switches, and each pixel consists of an acquisition unit and a processing unit.

In the algorithms mentioned above, the result of each pixel can be obtained by logic and arithmetic computations on the originally acquired signals of the pixels. They can be implemented for parallel processing. However, for an implementation in a pixel array circuit, the following issues should be addressed:

1. To place a large number of pixels in a limited semiconductor space for an acceptable resolution, the area for each pixel should be minimized.
2. It would be desirable not to have arithmetic operations in the thinning process, as they usually need space-consuming complex digital blocks. One may use analog modules for such operations. However, the pixels require certain homogeneity of characteristics, which is difficult to achieve in analog circuits due to technological process variations.
3. The inter-pixel data transfer for computation is a particularly critical issue in the design of the thinning process, as it is related to the number of inter-pixel connections to be implemented in the pixel array.

The input for thinning process is a binary image, and it is thus reasonable to develop a thinning process involving only simple logic operations and inter-pixel data transfer. Such a process is described in the following section.

2. PROCEDURE OF LOGIC OPERATIONS FOR PARALLEL THINNING PROCESS

The objective of a thinning process is to thin one kind of regions, such as ridge regions in a fingerprint image, as much as possible while preserving the original connectedness. The region pixels are assumed to have a value of logic 1 and background pixels of logic 0. In a thinning process, every pixel is examined with respect to its neighbors, and if it satisfies certain conditions not to be a pixel constituting the skeleton of its region, its value will be deleted, i.e. reset to 0.

The proposed thinning process is based on the same principle reported in [2] and [3]. It involves the examinations of the pixel values and the conditional deletions to grow the background regions as much as possible. However, unlike the algorithms in [2] and [3], the procedure in the proposed thinning process contains only simple AND, OR and XOR functions without requiring complex functional blocks for arithmetic operations.

In the process, like the others, a 3×3 operation window is used, as shown in Fig. 2(a). The conditions are based on the results of the comparisons of pixel values, by means of XOR operations as illustrated in Fig. 2(b). The comparisons are performed only with the vertically and horizontally adjacent pairs, excluding those in the diagonal directions. Thus, in the array circuit implementing this thinning process, diagonal inter-pixel connections will not be needed. However, the signals from the pixels located in the four corners of the windows are not ignored, as they are used to generate the comparison results at the North (N), South (S), East (E) and West (W) neighbors of the center pixel and these results are involved in the decision of deletion. Hence, the exclusion of the direct diagonal inter-pixel connections does not affect the quality of the processing.

In this thinning process, during each cycle of operation, i.e. an iteration, a background region may grow one-pixel further at its edge. To complete the thinning process of an image, several iterations may be needed to obtain the thinnest skeletons in the image. The number of iterations is related to the thickness of the thickest regions of the original binary image. Each iteration consists of two sub-iterations, in each of which different conditions are examined to determine the value of the center pixel *P*.

The conditions used in the two sub-iterations are as follows:

(a) $(X_1 \cdot X_2) + (X_2 \cdot X_3) + ... (X_7 \cdot X_8) + (X_8 \cdot X_1) = 0$

(b) $(X_9 \oplus X_{11}) + (X_{10} \oplus X_{12}) \neq 0$

(c1) $X_{10} + X_{11} + (X_9 \cdot X_{12}) = 0$

(c2) $X_9 + X_{12} + (X_{10} \cdot X_{11}) = 0$

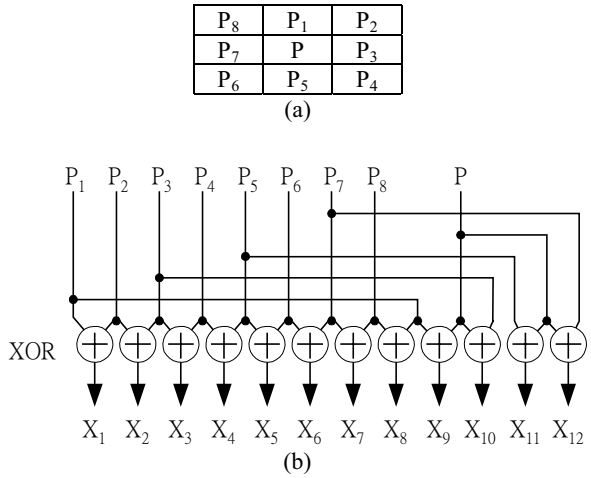

Fig. 2 (a) A 3×3 operation window.
(b) XOR operations for comparisons.

where the variables X_1, X_2, ... X_{11} and X_{12}, are the results of the XOR operations as indicated in Fig. 2(b).

If at least one of the terms in the left side of the logic expression is equal to 1, the condition (a) is not satisfied. That indicates that at least one of the pixels, P_1, P_2, ... P_7 and P_8, is different from its two adjacent neighbors that have the same value. Thus the center pixel *P* might be at the end or middle of a skeleton, the deletion should not take place at this stage. However, if the condition (a) is satisfied, it is still possible that the center pixel is part of a skeleton. There are two particular cases in which the condition (a) is satisfied, but the center pixel *P* should not be deleted to 0. The first case includes the pixel patterns shown in Fig. 3, where the center pixel is likely to be part of the skeleton. In the other case, all the pixels, P_1, P_2... P_8 and *P*, are 1, indicating that the center pixel *P* is not at the edge of its region. As this thinning process is, in fact, based on background region growing, the pixel value should remain unchanged during that sub-iteration. Conditions (b) are added to detect these two cases. Conditions (c1) and (c2) are introduced so that only a subset of pixels are qualified for deletion in each sub-iteration [3] to prevent wrong deletion in case of rectangles with two-pixel width [6].

As mentioned above, one iteration, i.e. one cycle of operation, consists of two sub-iterations. In the first sub-iteration, conditions (a), (b) and (c1) are examined, if all of them are satisfied, the center pixel will be deleted to 0. Otherwise the decision to delete it or not will be made in the second sub-iteration in which the comparisons will be

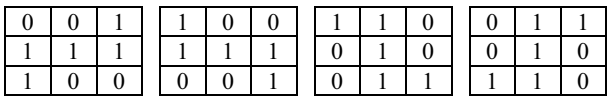

Fig. 3 Pixel patterns satisfying condition (a) but not (b).

redone and conditions (a), (b) and (c2) are checked. The center pixel *P* will be deleted to 0 if all these conditions are satisfied.

The proposed thinning process permits to implement circuit using only simple logic gates and small number of interconnections. Its efficiency is verified by the simulation results described in Section 3.

3. SIMULATION RESULTS

In order to verify the efficiency of the proposed thinning process, we simulated it with different image patterns. Fig. 5(a) is a binary fingerprint image, and Fig. 5(a') is the result of the proposed thinning process. Examining these two images, we can find that the processed image (a') preserves the connectedness of its original image (a). Moreover, all the ending pixels in Fig. 5(a') have only one ridge neighbor, which meets the requirement of the minutia detection for identification.

Simulations with images of different patterns have also been conducted. Fig. 5(b) is an image having thick regions combined with fine lines, (c) and (d) are text characters, and the processed images of these patterns are shown in (b'), (c') and (d') respectively. All the pixels that are not part of the skeleton of the region are deleted without breaking the connectedness of the region. Therefore, the proposed thinning process is suitable and efficient for different patterns in image processing.

4. CIRCUIT STRUCTURE

A pixel array circuit is proposed as an example of implementing the thinning process described in Section 2. A simplified structure of the circuit is shown in Fig. 4. In this structure, each pixel, indicated with a frame, has only four inter-pixel connections for data transfer, and it consists of an optical acquisition unit, switches, simple XOR and other logic gates and memories. The four connections *N*, *E*, *S*, and *W*, to the four neighboring pixels are bi-directional, and activated by a sequence of control. Each of the connections is used for transferring signals between two adjacent pixels. These signals can be acquired at the pixels or resulting from logic operations.

As mentioned in Section 2, the decision of deletion results from the logic operations on signals X_1, X_2, ... X_{11} and X_{12} that are generated in the XOR operations. In the pixel array, an XOR gate is inserted between every two adjacent pixels, as shown in Fig. 4, in horizontal or vertical directions. To minimize the circuit area and the power dissipation, Pass Transistor Logic (PTL) technique is used in the design. Each pixel consists of 115 MOS transistors, including those for the interconnection control, and most of them are minimum-sized. Using a currently available CMOS technology, we will be able to integrate it in a tiny space of 50×50 μm^2. Therefore the proposed thinning process can be implemented in a chip of pixel array with an acceptable resolution.

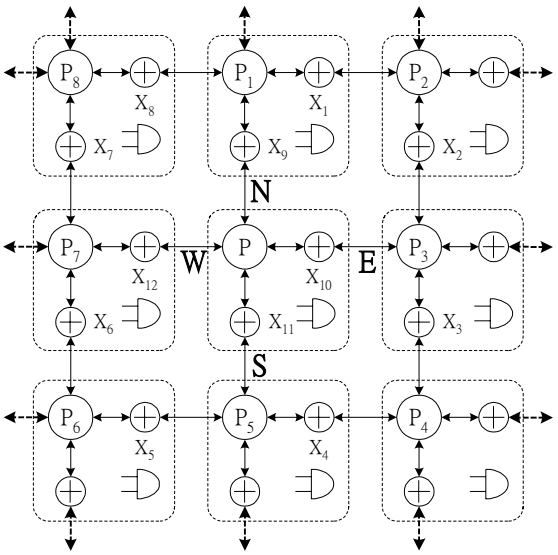

Fig. 4 Simplified structure of the pixel array for fingerprint image acquisition and thinning process.

5. CONCLUSION

In this paper, a thinning process suitable for pixel array implementation for parallel processing has been proposed. In this process, only simple logic operations are required, and these operations are based on the results of XOR operations of the adjacent pixels in horizontal and vertical directions. Simulation results demonstrated that the proposed thinning process is efficient for images with different patterns, and the quality of the thinned regions meets the requirement of a further processing such as minutia detection for personal-identification. The structure of a pixel array circuit is also designed as an example of implementation of the proposed thinning process. This array has simple inter-pixel connections and pixel circuits, and can be integrated easily in digital signal processing systems.

6. REFERENCES

[1] Anil K. Jain, Lin Hong, Sharath Pankanti and Ruud Bolle, "An Identity-Authentication System Using Fingerprints," *Proc. IEEE*, vol. 85, pp. 1365-1388, Sep. 1997.

[2] D. Rutovitz, "Pattern recognition," *J. Royal Statist. Soc. Series A*, vol. 129, no. 4, pp. 504-530, 1966.

[3] T. Y. Zhang and C. Y. Suen, "A Fast Parallel Algorithm for Thinning Digital Patterns," *Commun. ACM*, vol. 27, no. 3, pp. 236-239, Mar. 1984.

[4] H. E. Lu and P. S. P. Wang, "A comment on "A Fast Parallel Algorithm for Thinning Digital Patterns," *Commun. ACM*, vol. 19, no. 3, pp. 239-242, Mar. 1986.

[5] Jun-Sik Kwon, Jun-Woong Gi, and Eung-Kwan Kang, "An Enhanced Thinning Algorithm Using Parallel Processing," *Image Processing, 2001. Proceedings. 2001 International Conference on*, vol. 3, pp. 752 –755, 2001.

[6] L. Lam, S. W. Lee and C. Y. Suen, "Thinning methodologies- a comprehensive survey," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 14, no. 9, pp. 869-885, Sep. 1992.

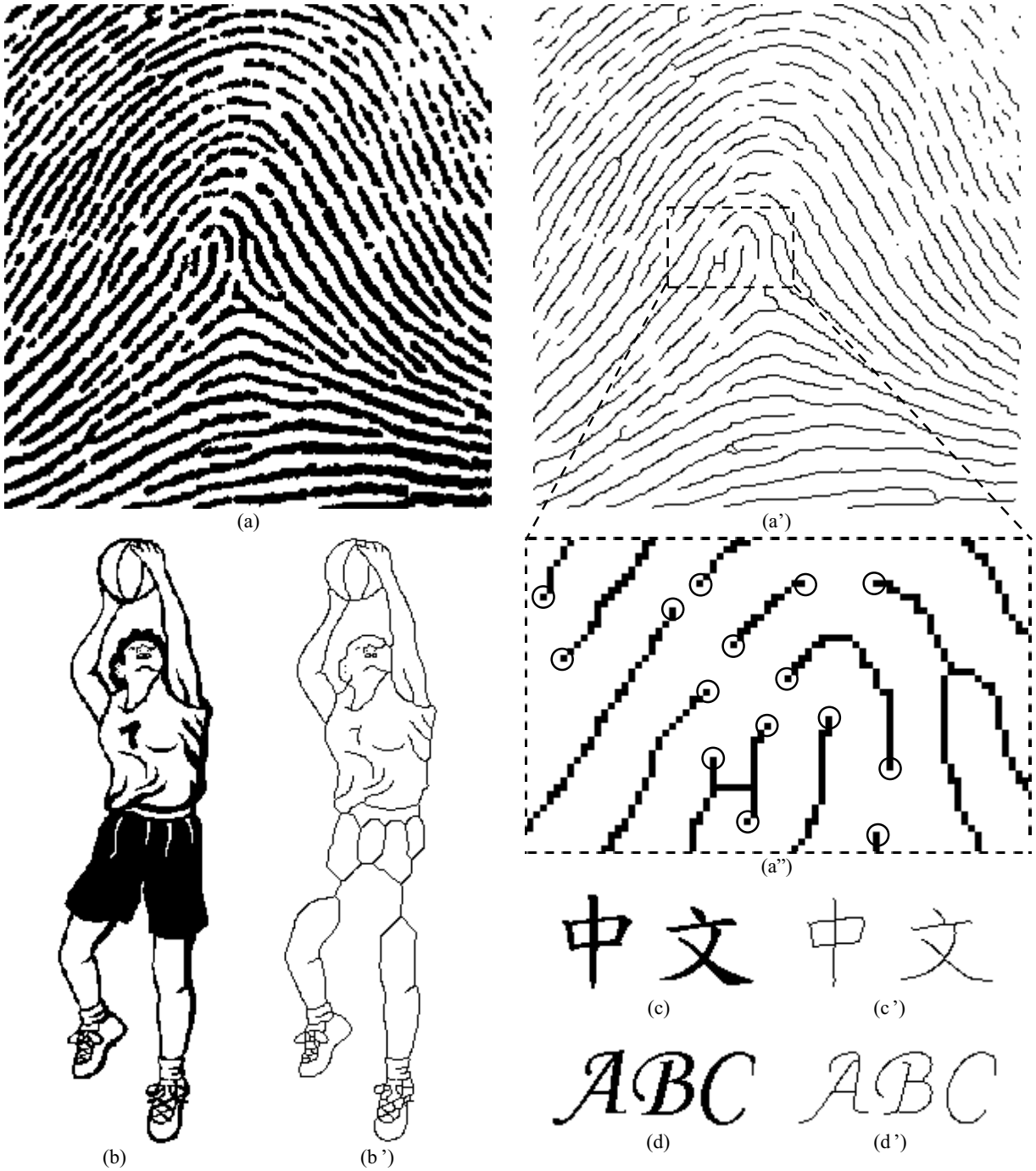

Fig. 5 (a) – (d) Original binary images.
(a') – (d') Corresponding results after the thinning process.
(a") Enlarged image of the region framed in (a'). It shows all skeletons with one-pixel-width at their ends, marked with small circles. All the ending pixels meet the requirement of the ridge ending as they have only one ridge neighbour in their 3×3 windows, which is required to facilitate the minutia detection for identification.