



Implementation of an Image Thinning Algorithm using Verilog and MATLAB

Dr G Sankar

Associate Professor, Department of CSE
Sri Sai Institute of Technology and Science, Rayachoti
Email: sankarphd@gmail.com

Abstract— Image thinning is a signal transformation that converts a thick digital image into a thin digital image or obtains its skeleton form. The skeleton expresses the structural connectivities of the main component of a object and is one pixel in width. Skeletonization reduces the original image into a more compact representation. A basic method of skeletonization is thinning. In this paper image thinning operation has been implemented on a binary image of 128 x 128 pixels using Zhang Suen's thinning algorithm. The proposed work is designed using MATLAB 7.12 and synthesized by mapping on Virtex 5 in Xilinx ISE for understanding the hardware complexity. Simulation results are obtained in terms of waveforms in ISim Xilinx ISE Simulator and the output text file of the hardware system is converted to an image format using MATLAB.

Keywords – Image thinning, skeletonization, Zhang – suen's algorithm, digital image processing, Verilog, MATLAB, Hardware Description Language (HDL), iteration.

I. INTRODUCTION

Thinning is a process of extracting a skeleton from an object in a digital image. It can also be defined as act of identifying those pixels belonging to an object that are essential for communicating the object's shape: these are the skeletal pixels, and form a set. Thinning provides a convenient and condensed representation of image object information. Skeleton of an object can preserve topological properties, reduce storage requirements and reduce the transmission time. Thinning is also termed as skeletonization. Skeletonization is widely used in many image pre-processing applications, such as character recognition, pattern recognition, image coding and biological shape description. The proposed work focuses on extracting the centre line of a binary image using Zhang-Suen thinning algorithm.

A Overview of Skeletonization

Skeletonization or thinning operation is done by iteratively eroding away nonessential pixels from the image until one is left with a stick figure, or skeleton, which describes the object. Skeletonization algorithms are the need to compute a reduced amount of data or to simplify the shape of an object in order to find features for recognition algorithms and classifications. Additionally the transformation of a component into an image showing essential characteristics can eliminate local noise at the frontier.

The result of skeletonization using thinning algorithms must have the following properties:

- As thin as possible
- Connected
- Centred

All the thinning algorithms are classified into two broad categories:

- 1) Iterative thinning algorithm
- 2) Non iterative thinning algorithm

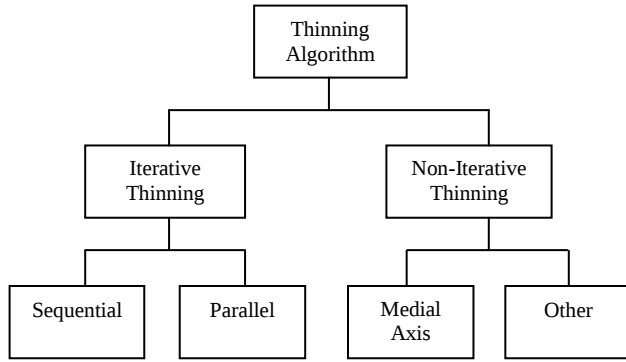


Figure 1: Classification of thinning algorithms

In iterative methods, thinning algorithms produce a skeleton by examining and deleting contour pixels through an iterative process in either sequential or parallel way. Sequential thinning algorithms which examine contour points of an object in a predetermined order, and this can be accomplished by either raster scanning or following the image by contour pixels. In parallel thinning algorithms, pixels are examined for deletion on the basis of results obtained only from the previous iteration. Hence parallel thinning algorithms are suitable for implementation in parallel processors.

Non-iterative (non pixel based) thinning is not based on examining individual pixels. These algorithms produce a certain median or centre line of the pattern to be thinned directly in one pass, without examining all the individual pixels. Some popular non pixel based methods include medial axis transforms, distance transforms, and determination of centrelines by line following. Medial axis transforms often use gray-level images where pixel intensity represents distance to the boundary of the object. Distance transform based methods compute the distance to the image background for each object pixel and use this information to determine which pixels are part of the skeleton.

B. Zhang-Suen's Thinning Algorithm

The Zhang-Suen's thinning algorithm is an iterative algorithm. It is fast and simple to be implemented. In this algorithm it is assumed that the region point in the image has pixel value '1' and background points have value '0'. The Zhang Suen's method consist of successive passes of two basic steps applied to the contour points of the given region, where a contour point is any pixel with value '1' and having at least One 8-neighbor valued '0'. With reference to the 8-neighborhood definition shown in figure 1, the first step flags a contour point p for deletion if the following conditions are satisfied:

- (a) $2 \leq N(P1) \leq 6$.
- (b) $S(P1) = 1$.
- (c) $P2 * P4 * P6 = 0$.
- (d) $P4 * P6 * P8 = 0$.

where $N(p1)$ is the number of nonzero neighbors of $p1$;
that is,

$$N(P1) = P2 + P3 + \dots + P8 + P9$$

and $S(p1)$ is the number of 0-1 transition in the ordered sequence of $p2, p3, \dots, p8, p9$. For example, $N(p1) = 4$ and $S(p1) = 3$ in figure 2.

P9	P2	P3
P8	P1	P4



In the second step, conditions (a) and (b) remain the same, but conditions (c) and (d) are changed to:

- (c') $P2 * P4 * P8 = 0$.
- (d') $P2 * P6 * P8 = 0$.

Step 1 is applied to every border pixel in the binary region under consideration. If one or more of the conditions (a) through (d) area violated, the value of the point in question is not changed. If all conditions area satisfied the point is flagged for deletion. It is important to be considered, that the point is not deleted until all border points have been processed. This prevents changing the structure of the data during execution of the algorithm. After step 1 has been applied to all border points, those that were flagged are deleted, changed to '0' for example. Then, step 2 is applied to the resulting data in exactly the same manner as step 1.

II. LITERATURE SURVEY

Thinning is a process of reducing a shape to a simpler that still retains the essential features of the original object [1]. A common use of thinning is in the pre-processing stage to facilitate higher level analysis and recognition for applications such as Optical Character Recognition, diagram understanding, fingerprint analysis, and feature detection for computer vision. In [2] an improved fast thinning algorithm is proposed to accelerate the process for thinning fingerprint images. The algorithm is implemented both in Matlab and on FPGA using VHDL. In [3] authors show skeletonization of an binary image in which they use gradient and watershed algorithm to extract what can be an image in actual and perform the skeletonization over the image to avoid higher intensity values. In [4] compares root mean square errors on skeletonization using Mathematical Morphology and Voronoi Diagrams. And the authors found that best accuracy of skeletonization is by Mathematical Morphology.

In [5] the author describes that hardware implementation of thinning algorithm is bit difficult because data is stored in memory with sequential/serial access on it when compared with high level language accessing image pixel value stored in matrix is easy by using index of the matrix for example P(row,col). In [6] the author presents a remedy to overcome the drawbacks of traditional thinning algorithm like low speed, deformation by a novel approach which generates the skeletons of binary patterns via block decomposition and contour vector matching.

III. METHODOLOGY

The entire code for thinning of binary image of size 128 x 128 is developed in Xilinx 12.1 using verilog, a hardware descriptive language and in MATLAB 7.12.

A. Functional Block Diagram

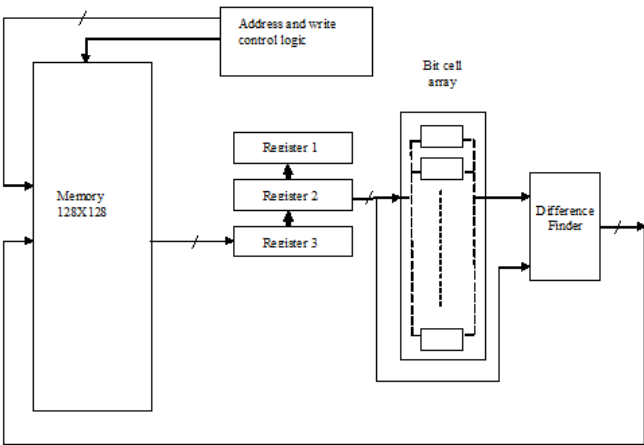


Figure 3: Functional Block Diagram

Figure 3 shows the functional block diagram for Zhang – Suen’s thinning algorithm using Verilog. From the block diagram it is clear that, the binary image of size 128 x 128 pixels is stored in the internal memory. To access the information, RAM must be addressed to get the values of the pixels that are required for processing. Therefore address and write control logic is required, which sequentially access the information stored in internal memory. Read the frames of data that need to be processed repeatedly till the converge condition is met. That is, the 9 bit matrix arranged as [p1,p2,p3...p9] is fed to the bit cell array. The data obtained is only one bit p1(either 0 or 1). The bit pattern is computed and the three conditions are evaluated. Accordingly p1 is set or reset. The process is repeated till the counter becomes zero according to the algorithm.

B. Flowchart

The flowchart shows the details of the image thinning algorithm. The number of iterations depends on the binary image being thinned.

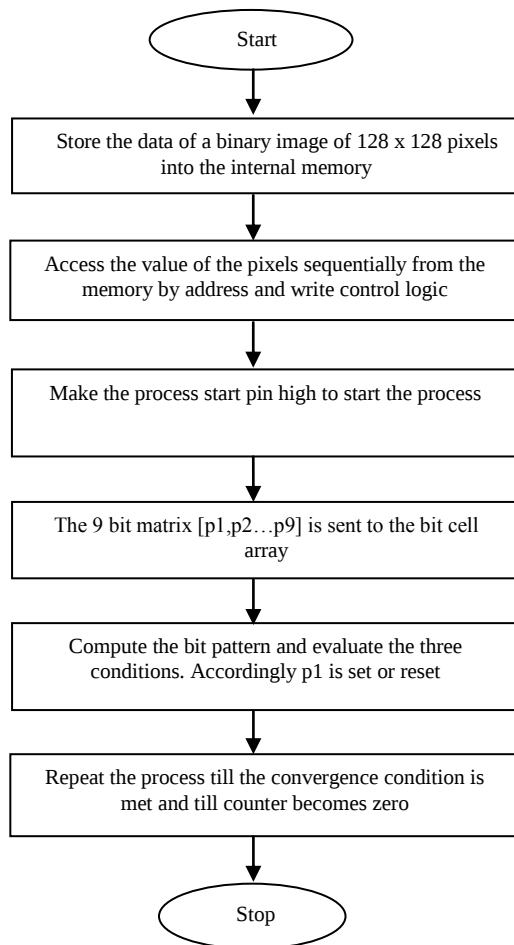
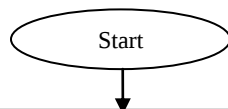


Figure 4: Procedure for image thinning using Verilog



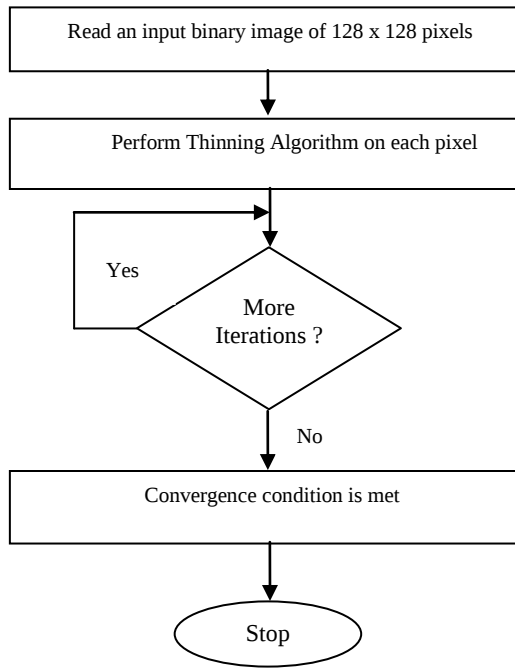


Figure 5: Procedure for image thinning using MATLAB

Figure 4 and figure 5 shows the flowchart design for image thinning using Verilog and MATLAB respectively.

IV. RESULTS

A. Verilog Results

In this section, experimental results are demonstrated. The image thinning algorithm is developed in Xilinx ISE 12.1. It is designed using Verilog by storing the binary image in the memory. The pixel values stored in the memory are accessed with the help of address and write control logic. Then each and every pixel is examined and contour pixels are deleted through an iterative process, to produce a skeleton of an image.

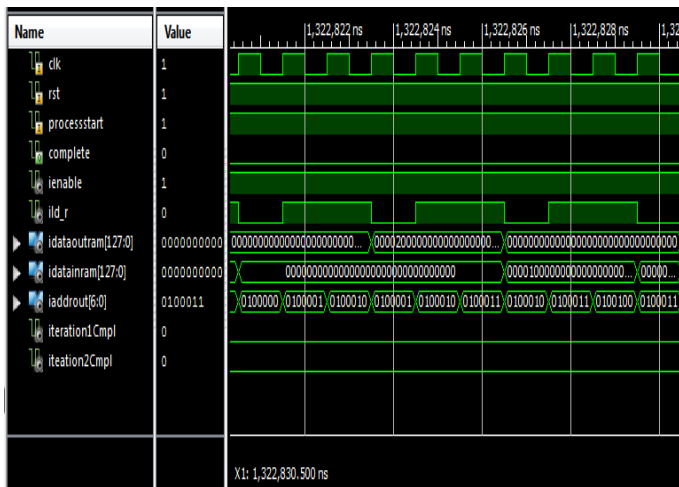


Figure 6: Simulation results of image thinning

Figure 6 shows the simulation results of image thinning. Here input image matrix is in a binary value. In graph `idatainram [127:0]` represents the original image matrix of size 128 x 128 and `idatoutram[127:0]` represents the image matrix after thinning.

[illegible]

Figure 7: Memory file of an unprocessed image

[illegible]

Figure 8: Memory file of a processed image

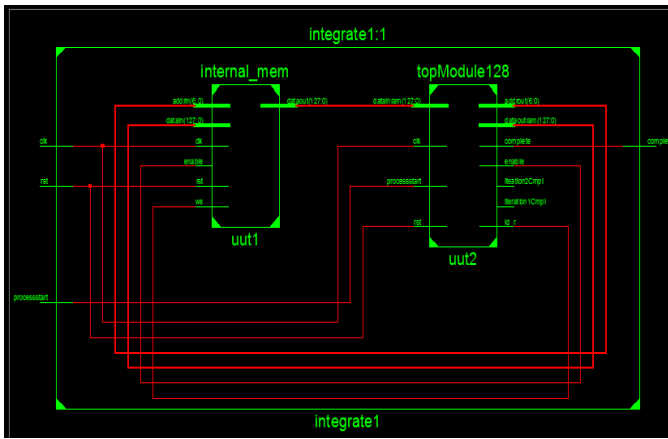


Figure 9: RTL schematics of the internal memory and data processing unit

Figure 7 shows a memory file of an unprocessed binary image. The zeros and ones represent the values of pixels stored sequentially in the memory. The background points have value '0' which appear black in color and the region point in the image have pixel value '1'. Figure 8 is the memory file of a processed binary image. It clearly indicates the thinning of a binary image i.e the pixels satisfying the conditions has mentioned in the proposed algorithm is set '0' yielding skeletonization of the region. The RTL schematics of the internal memory and data processing unit are generated in Xilinx ISE tool as shown in the Figure 9.

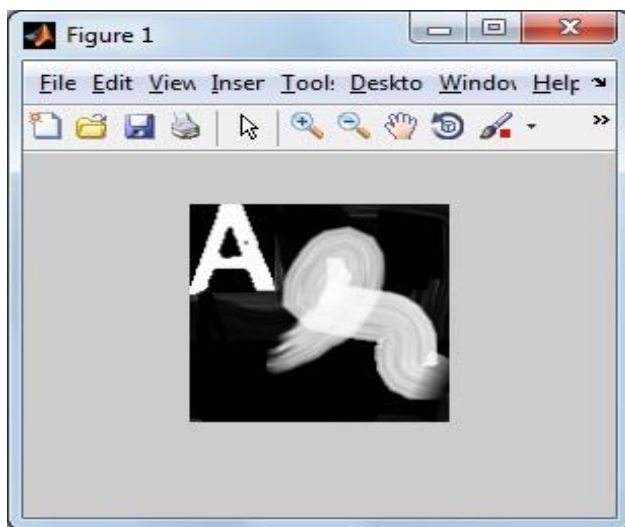


Figure 10: Original image before thinning

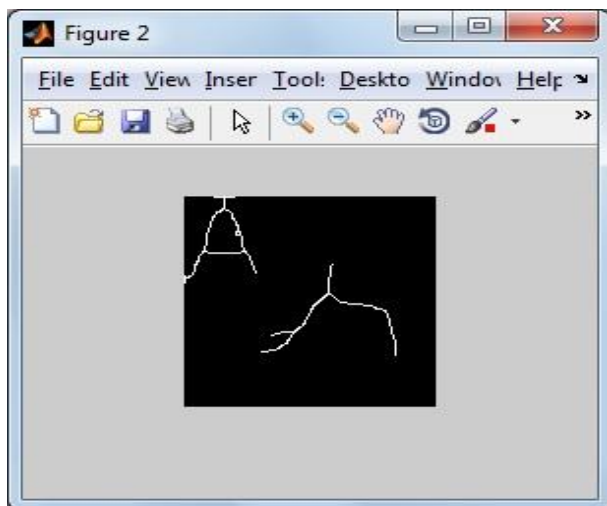


Figure 11: Image after thinning obtained from Verilog

Figure 10 is the original image before thinning. This image is obtained by converting an input text file to image format using MATLAB. The verilog output results a output text file with processed pixel values obtained by Xilinx's tool. This text file is then converted to image format using MATLAB which shown in figure 11.

B. MATLAB Results

The Zhang – Suen's image thinning algorithm is also developed using MATLAB 7.12. The mushroom image of size 128 x 128 is considered as the input image shown in figure 12. The thinned image of mushroom obtained from the MATLAB is shown in figure 13.

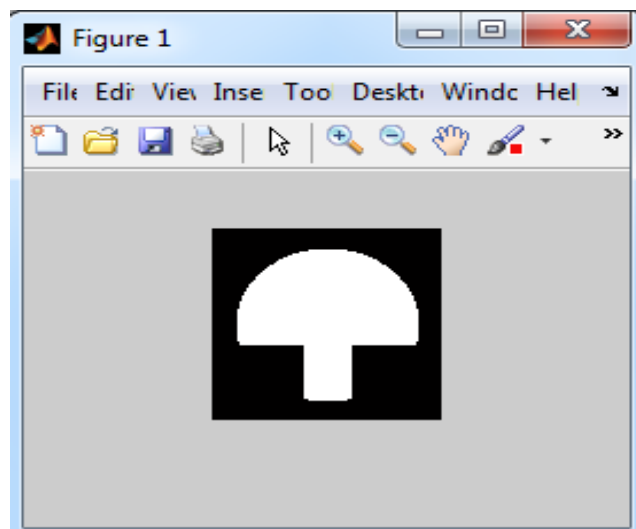


Figure 12: Original image before thinning

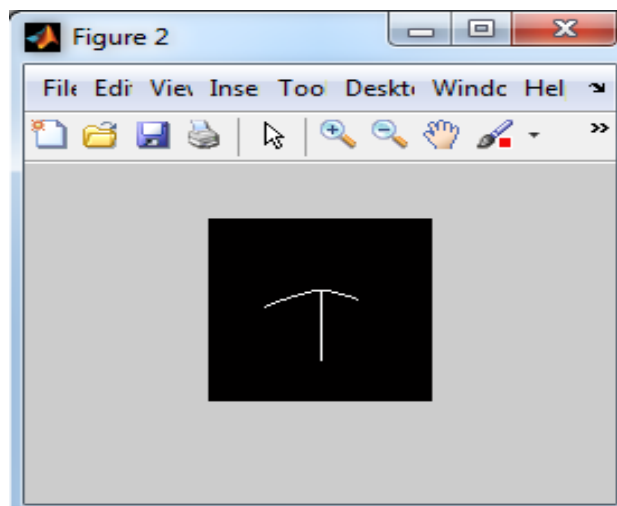


Figure 13: Image after thinning obtained from MATLAB

The Zhang – Suen’s image thinning algorithm is also developed using MATLAB 7.12. The mushroom image of size 128 x 128 is considered as the input image shown in figure 12. The thinned image of mushroom obtained from the MATLAB is shown in figure 13.

V. CONCLUSION

The pixel based thinning algorithm is designed and simulated using Verilog and MATLAB. Simulation results obtained from verilog are shown through ISim Simulator Xilinx ISE 12.1. The proposed algorithm designed in Verilog is more efficient because the complete frame of an image is read at once there by reducing the iteration by 128 times. The results obtained from the MATLAB and Xilinx should be compared.

ACKNOWLEDGMENT

The authors thank the authorities of Sri Dharmasthala Manjunatheshwara College of Engineering and Technology, Dhavalagiri, Dharwad, Karnataka, India for encouraging them for this research work.

REFERENCES

- [1] J. Serra, “*Image Analysis and Mathematical Morphology I II*” Academic Press, 1982.
- [2] Gayathri S and Dr V Sridhar, “*An Improved Fast Thinning Algorithm for Fingerprint Image* ” International Journal of Engineering Science and Innovative Technology, Volume 2, Issue 1- January 2013.
- [3] Kamaljeet Kaur and Mukesh Sharma,” *A Method for Binary Image Thinning using Gradient and Watershed Algorithm*” International Journal of Advanced Research in Computer Science and Software Engineering, Volume 3, Issue 1- January 2013.
- [4] Neeti Daryal and Vinod Kumar, “*A Comparison of Root Mean Square Errors on Skeletonization Methods*” International Journal of Applied Physics and Mathematics, Vol. 2, No. 1, January 2012.
- [5] Lingga Hermanto, Sunny Arief Sudiro and Eri Prasetyo Wibowo, “*Hardware Implementation of Fingerprint Image Thinning algorithm in FPGA Device*” International Conference on Networking and Information Technology 2010.
- [6] Kuo-Chin Fan, Den-Fong Chen and Ming-Gang Wen, “*Skeletonization of Binary Images with Non uniform Width Via Block Decomposition and Contour Vector Matching*” Pattern Recognition, Vol. 31, No. 7, pp. 823Ð838, 1998.
- [7] Alberto Martin and Sabri Tosunoglu, “*Image Processing Techniques for machine vision*”.