Research Journal of Applied Sciences, Engineering and Technology 6(24): 4637-4641, 2013
DOI:10.19026/rjaset.6.3479
ISSN: 2040-7459; e-ISSN: 2040-7467

Submitted: February 27, 2017 Accepted: April 29, 2013 Published: December 25, 2013

Research Article

Adaptive Thresholding based Image Binarization Using VHDL

[1]S. Swaminathan, [1]A. Srinivasan, [2]D. Narasimhan and [1]K. Ramasamy
[1]Department of ECE,
[2]Department of Mathematics, Srinivasa Ramanujan Centre,
SASTRA University, Kumbakonam, India

Abstract: The aim of this study is to recognize the given image with the existing image based on the technique of image binarization by MATLAB tool and it is simulated using VHDL (Very High Speed IC Hardware Descriptive Language) using MODELSIM tool. This image binarization is based on LEGION (Locally Excitatory Globally inhibitory Oscillatory Network) Concept. In this study the algorithm finds an optimum threshold technique, the other by separating the image background and foreground pixels. This algorithm has superior performance in separating the images from background in comparison with the other threshold techniques.

Keywords: Adaptive thereshold, binarisation, MATLAB, modelsim, VHDL

INTRODUCTION

To extract the required image from a given image, it may be separated into different components say foreground and background. An adaptive threshold concerning extraction of image from the background in a given image sequences is possible for implementation in hardware. The conventional histogram based threshold methods are deficient in detecting images due to poor contrasts between image and the background, or the change of illuminations (Otsu, 1979; Rosenfeld and De La Toree, 1983; Sezan, 1985; Trier and Taxt, 1995). But in real time applications, except adaptive thereshold techniques are not fast enough for hardware implementation (Kittler and Illingworth, 1986). The image binarization (Lee *et al*., 1991; White and Rohrer, 1983) is a process of dividing the original image into two components, foreground or image and background. The gray level value of background (value '0') is different from the foreground (value '255') values. So, for the effective separation of foreground and background, this thresholding techniques may be considered (Hertz and Schafer, 1988; Sezign and Sankar, 2004). The theresholding techniques yields a binary image as an output by assigning pixels with values less than the threshold as 0's and remaining pixels as 1's. In this study the Image binarization is carried out by MATLAB tool and it is simulated using VHDL (Very High Speed IC Hardware Descriptive Language) using MODELSIM tool (XESS Corporation, 2001). This image binarization is based on LEGION (Locally Excitatory Globally inhibitory Oscillatory Network) Concept (Ligon *et al*., 1998).

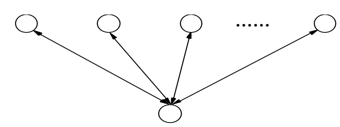

Fig. 1: Comparator model

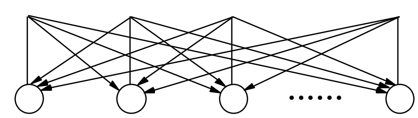

Fig. 2: Globally connected

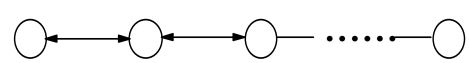

Fig. 3: Locally connected

The LEGION concept has the following features:

- Binarization is realized in a very short time due to the simple processing.
- Due to the simple structure of each cell representing a pixel compact integration of many cells on a single chip becomes possible.
- Fast software implementation is possible.

Neural oscillator concept: There are three possible ways to reach synchrony as shown in Fig. 1 to 3.

LEGION is composed of a Basic Oscillator, Local Excitatory Connections (produce phase synchrony within each object) and Global Inhibitor (it receives inputs from the entire network and feeds back with

Corresponding Author: S. Swaminathan, Department of ECE, Srinivasa Ramanujan Centre, SASTRA University, Kumbakonam, India

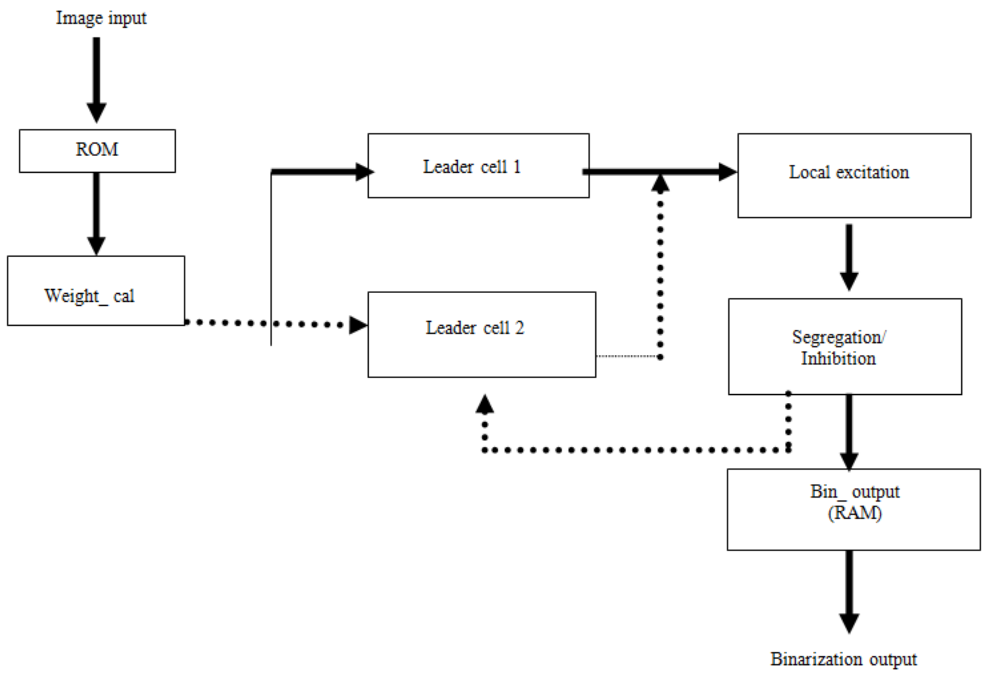

Fig. 4: Block diagram

inhibition). Each Oscillator will respond to a detected feature at some location in the image. Synchronized group of oscillators is separated from other synchronized groups that represent different objects by de-synchronization. It is proposed in Digital implementation, Model described by digital algorithm, both gray/color scale binarization and segmentation in a single algorithm (Yanowitz and Bruckstein, 1989). Re-programmability is possible using FPGA (Shirazi *et al*., 1995).

The Fig. 4 shows Digital LEGION Image Binarization Architecture. It consists of six blocks namely:

- Input ROM
- Weight calculation
- Leader cell
- Locally excitatory
- Segregation/Inhibition
- Output RAM

DIGITAL LEGION ALGORITHM

Initialization:

Step 1: Initial global inhibitor, Z = 0
Step 2: Calculation of Weights

Gray scale images:

$$W_{ij} = \frac{I\ max}{1 + | I_i - Ij |}$$

where,
W_{ij} = Weight value between i^{th} and j^{th} pixels
I_i = I^{th} pixel value
I_j = J^{th} pixel value
I_{max} = Max. Gray level value $[2^n - 1]$

Color image:

$$W(R)_{ij} = \frac{I(R)\ max}{I + | (IR)\ I - (IR)_j |}$$

$$W(G)_{ij} = \frac{I(G)\ max}{I + | (IG)\ I - (IG)_j |}$$

$$W(B)_{ij} = \frac{I(B)\ max}{I + | (IB)\ I - (IB)_j |}$$

$$W_{ij} = \min \{W(R)_{ij}, W(G)_{ij}, W(B)_{ij}\}$$

Step 3: Detection of Leader Cell (P)

$$\sum_{j=1}^{N} \text{If } (\sum W_{ij} > Ti) \text{ then } Pi = 1, \text{ otherwise } Pi = 0$$

where,
W_{ij} = weight value between i^{th} pixel and j neighboring pixels
j = 1, 2, 3…N
Ti = Leader cell threshold value

Step 4: Set all cell to non-excitation $X_i(0) = 0$.

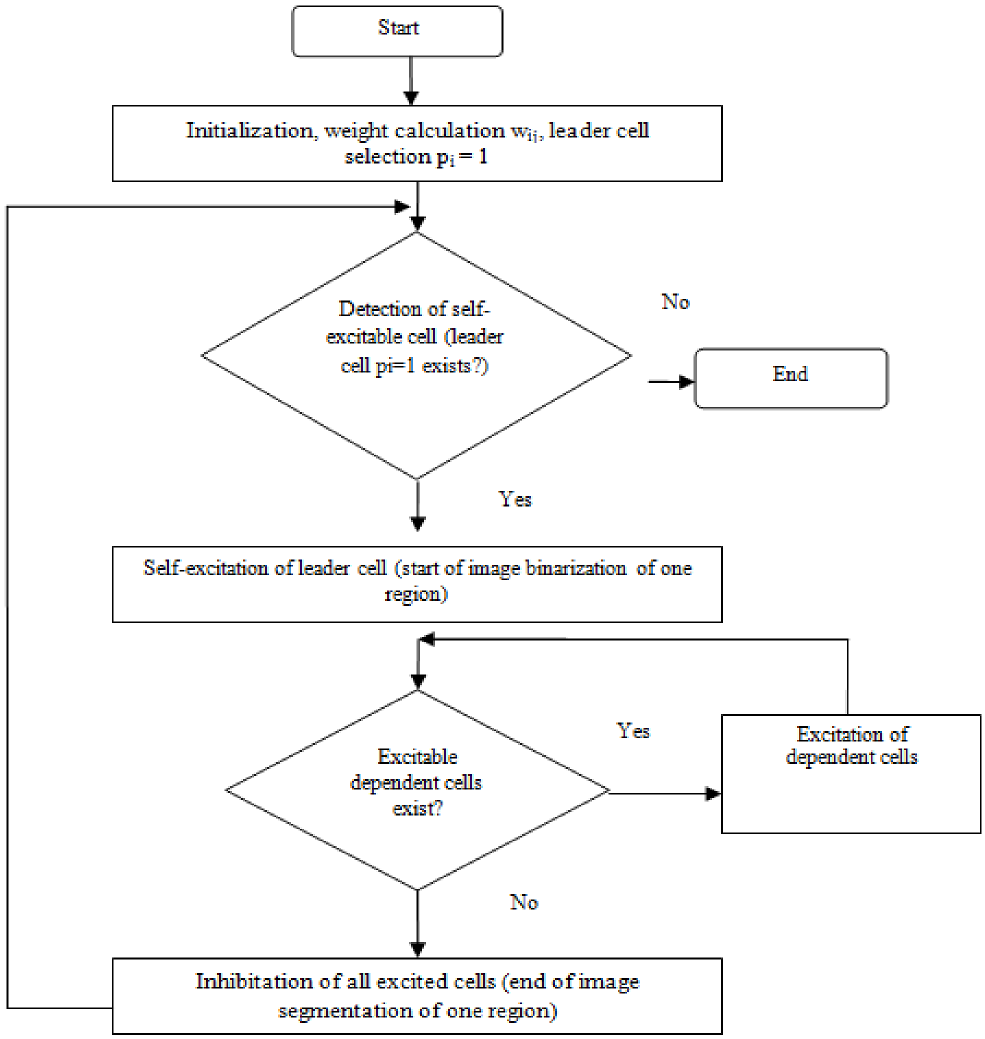

Fig. 5: Flow chart-LEGION algorithm

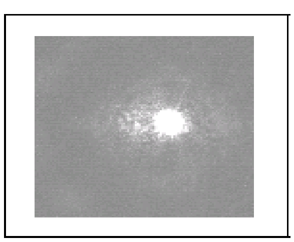

Fig. 6: Sample output

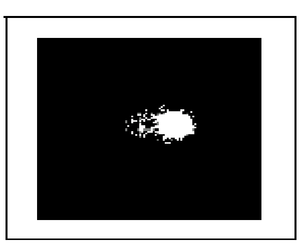

Fig. 7: Binarized image

Self-excitation of leader cell: If (excitable cells = false) then stop; //terminate Otherwise (find leader P_i = 1) then

$X_i = 1$, $Z_i = 1$; goto (step 3) //Self-Excitation Otherwise (step2);

Identification of dependent cell (or) local excitation: If pixel Ii is a neighbor of $X_i = 1$ and $WI_iPi>T2$ then Ii is a dependent cell of X_i.

Where, WI_iP_i = weight between pixel I_i and weight corresponding to $X_i = 1$ location.

Segregation:

- Identify first excitation pixel $X_i = 1$ and read corresponding pixel value I_i
- Identify $X_i + 1 = 1$ pixel, read corresponding pixel value $(I_i + 1)$
- If $I_i + 1 - I_i < T3$ then $X_i + 1 = 1$

Table 1: Input ROM

24	36	36	127	24
127	24	26	36	127
127	127	24	36	36
128	126	24	24	127
36	24	127	24	36

Table 2: Output of weight calculation block

276	550	786	263	23
516	573	1060	1060	263
767	773	826	826	533
344	476	1043	826	10
24	280	135	533	40

Table 3: Local threshold is 200. Output of a leader cell-pixel position

0	0	0	0	0
0	0	1	1	0
0	0	0	0	0
0	0	1	0	0
0	0	0	0	0

Table 4: Output of a local excite cell-position

1	1	1	0	0
0	1	1	1	0
0	0	1	1	1
0	0	1	1	0
0	1	0	1	0

- Keep I_i value as same for all other excitation pixel $X_i + 2$, $X_i + 3$ …$X_i + N$
- Repeat the process for all $X_i + N = 1$ pixels; N = 1, 2, 3…

Table 5: Output of leader cell 2-pixel position segregation threshold is 10

1	0	0	0	0
0	1	0	0	0
0	0	1	0	0
0	0	1	1	0
0	1	0	1	0

Table 6: Output of segregation block-pixel position

2	0	0	0	0
0	2	1	1	0
0	0	2	0	0
0	0	2	2	0
0	2	0	2	0

Inhibit all segregated output: First segmentation is over, if still more segmentation is there goto step 3 and repeat the algorithm. The flow chart (Fig. 5) is the description of LEGION algorithm.

RESULTS AND DISCUSSION

The following (Fig. 6 and 7) are the sample image and its binarization.

The sample image is given to MATLAB Tool and the Corresponding Matrix values are obtained. The following tables (Table 1 to 6) shows the output of the (Input ROM, weight calculation, leader cell 1 and 2, local excite and output of RAM respectively) in the digital LEGION image binarization architecture.

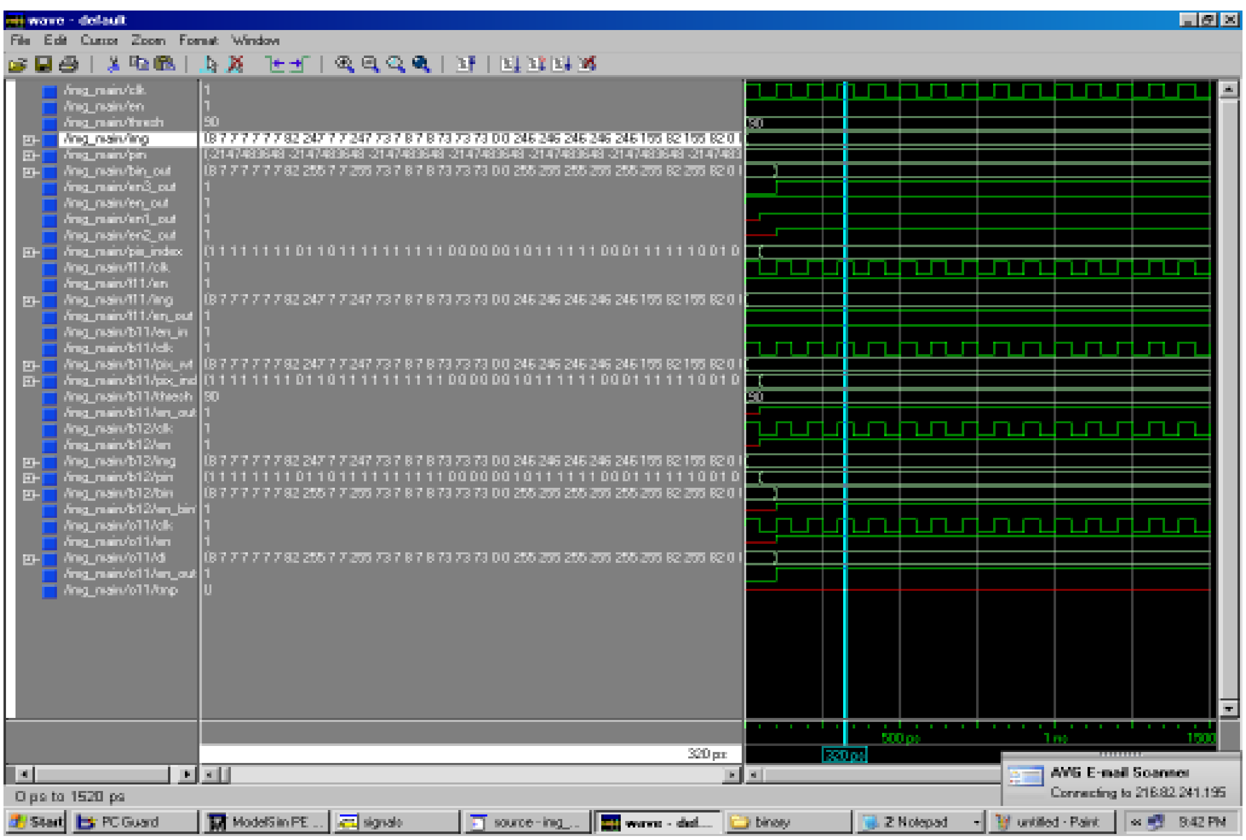

Fig. 8: VHDL simulated output

The weight value between 36-24 is 19, 127-36 is 2, 127-24 is 2 and 127-128is 127.
Let the leader threshold value be 1000.

The corresponding simulated output using VHDL is shown in Fig. 8.

CONCLUSION

In the present study, the image binarization is done by MATLAB and the same is simulated using VHDL by using MODELSIM. In this study the algorithm finds an optimum threshold technique, the other by separating the image background and foreground pixels. This algorithm has superior performance in separating the images from background in comparison with the other threshold techniques.

REFERENCES

Hertz, L. and R.W. Schafer, 1988. Multilevel thresholding using edge matching. Comp. Vision Graphic. Image Process., 44: 279-295.

Kittler, J. and J. Illingworth, 1986. Minimum error thresholding. Pattern Recogn., 19: 41-47.

Lee, S.W., L. Lam and C.Y. Suen, 1991. Performance evaluation of skeletonizing algorithms for document image processing. Proceeding of the 1st International Conference Document Analysis and Recognition, pp: 260-271.

Ligon, W.B., S. McMillan, G. Monn, K. Schoonover, F. Stivers and K.D. Underwood, 1998. A Re-evaluation of practically of floating point operations on FPGAs. Proceeding of the IEEE Symposium on FPGAs Custom Computing Machines, pp: 206-215.

Otsu, N., 1979. A threshold selection method from gray level histograms. IEEE T. Syst., Man Cyb., 9: 62-66.

Rosenfeld, A. and P. De La Toree, 1983. Histogram concavity analysis as an aid in threshold selection. IEEE T. Syst. Man Cyb., 13: 231-235.

Sezan, M.I., 1985. A peak detection algorithm and its application to histogram-based image data reduction. Graph. Mod. Image Process., 29: 47-59.

Sezign, M. and B. Sankar, 2004. Survey over image thresholding techniques and quantitative performance evaluation. J. Electron. Imaging, 13(1): 146-168.

Shirazi, N., A. Walters and P. Athanas, 1995. Quantitative analysis of floating-point arithmetic on FPGA based custom computing machines. Proceeding of the IEEE Symposium on FPGAs for Custom Computing Machines. California, April 1995.

Trier, O.D. and T. Taxt, 1995. Evaluation of binarization methods for document images. IEEE T. Pattern Anal., 17(3): 312-315.

White, J.M. and G.D. Rohrer, 1983. Image thresholding for optical character recognition and other applications requiring character image extraction. IBM J. Res. Dev., 27(4): 400-411.

XESS Corporation, 2001. Using Xilinx webPack Software to Create FPGA Designs for the XSA Board. Retrieved from: www.xess.com.

Yanowitz, S.D. and A.M. Bruckstein, 1989. A new method for image segmentation. Comput. Vision Graph., 46: 82-95.