

A Parallel VLSI Architecture for Real-Time Segmentation of Images with Complex Background Environment

Ming-Jung Seow and Vijayan K. Asari
Department of Electrical and Computer Engineering
Old Dominion University, Norfolk, VA 23529

Abstract – A parallel VLSI architecture for real-time segmentation of endoscopic images is presented in this paper. The architecture is based on a neural network implementation of an adaptive progressive thresholding (APT) algorithm that precisely segments the darkest regions of an endoscopic image representing the gastrointestinal lumen. The segmentation process is an extension of a comprehensive statistical technique based on linear discriminant analysis for partitioning the image. The APT algorithm is modeled as a recurrent neural network with adaptive weight modifications for rapid convergence. The hardware design utilizes simple processing elements functioning simultaneously to generate the optimum threshold for performing segmentation in real-time.

1 Introduction

The lumen region and boundary in intestinal images form the preliminary basis of the features used for navigation and guidance in an automated endoscopy system. The high speed and accurate extraction of these features is essential for real-time navigation. Since the endoscope uses several light sources at its tip and the illuminating distances of these sources are limited, the intestinal surface lying near the light source will be brighter than the farther ones. Hence, the areas of lowest intensity represent the lumen region in an image. Hardware realization of a novel technique based on adaptive progressive thresholding (APT) for the real-time segmentation of endoscopic images is presented in this paper. Adaptive thresholding is based on a discriminant analysis that partitions the image into two classes G_0 and G_1 at gray level 't' such that $G_0 = \{0, 1, 2, \dots, t\}$ and $G_1 = \{t+1, t+2, \dots, L-1\}$, where L is the total number of gray levels in the image. The optimum threshold t^* can be obtained by maximising the between-class variance. Hence,

$$t^* = \text{Arg Max}_{0 \leq i \leq L} \{\eta\} < \alpha \mu_T \quad (1)$$

where,

$$\eta = \sigma_B^2$$

$$\sigma_B^2 = w_0 w_1 (\mu_1 - \mu_0)^2$$

8.2.2

w_0 and w_1 denote the fraction of pixels lying in G_0 and G_1 class, respectively, and can be represented by,

$$w_0 = \sum_{i=0}^t n_i \quad w_1 = N - w_0 \quad (2)$$

where n_i is the number of pixels on the i th gray and N is the total number of pixel in the image. μ_0 and μ_1 represent the class means for G_0 and G_1 , respectively, and are calculated by,

$$\mu_0 = \frac{\mu_t}{w_0} \quad \mu_1 = \frac{\mu_T - \mu_t}{(1 - w_0)} \quad (3)$$

where,

$$\mu_t = \sum_{i=0}^t i \times n_i \quad \mu_T = \sum_{i=0}^{L-1} i \times n_i$$

Repeated application of this technique adaptively in a systematic fashion leads to a more accurate thresholded value. Recursive thresholding is continued until the Cumulative Limiting Factor (CLF) defined as the ratio of between-class variance and total variance in every Δ^{th} iteration becomes less than a predefined value. The algorithm involving in calculating the APT is shown on Fig. 1.

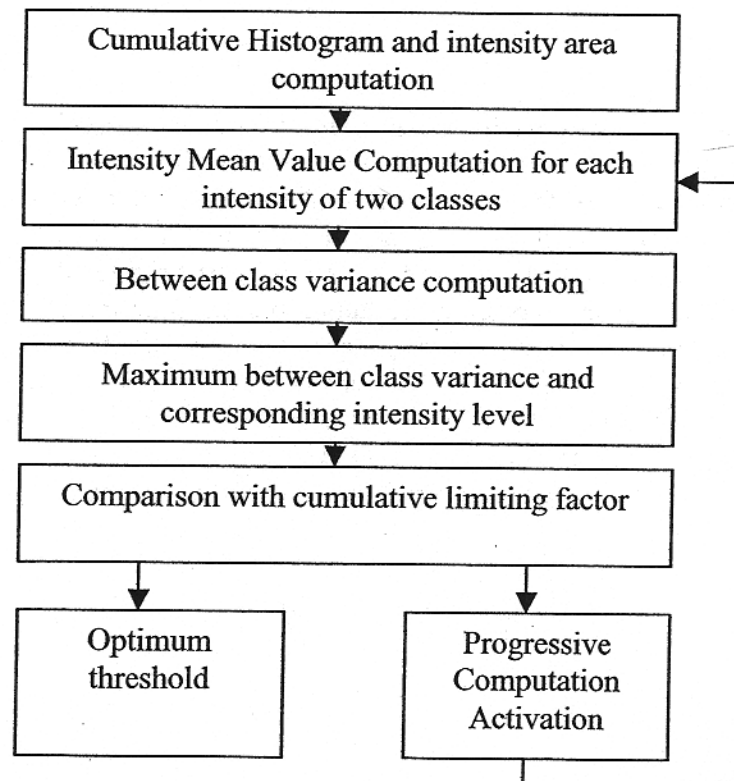


Fig.1 . Computational steps in APT

2 Parallel Architecture for APT

The neural architecture is a recurrent Neural-Network, which consists of 3 layers. Each hidden processing element is fully connected to all the input processing elements, and each output processing element is connected to itself and fully connected to other output processing elements, and all hidden processing elements. In addition, each output processing element is fully connected to the input processing elements. The parallel architecture for the APT is made for four-bit gray level image of size 32×32 . Some variations in the computational steps of the APT algorithm are adopted to make the hardware implementation feasible.

2.1. Architecture for the input layer

The input layer is simply a distribution layer where it takes the histogram values n_i , and forward it to the hidden layer. For example, the input will get the histogram intensities from the external input, it will store the sixteen intensity values into sixteen 10 bit registers for the next cycle, and it will forward the histogram values to the hidden layer. Furthermore, the input layer also accepts feedback loop from the output layer to activate or inhibit the inputs n_i by using a simple 15 bit AND gate for enabling the register. Fig. 2 illustrates 1 of the 16 elements in the input layer.

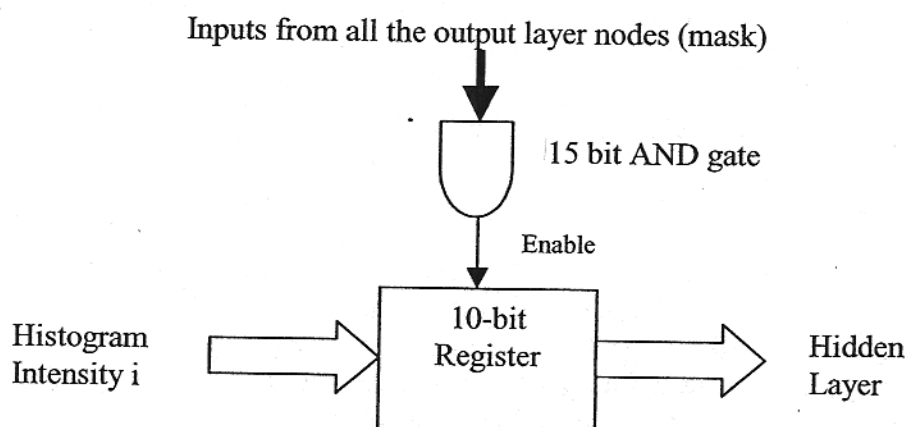


Fig. 2. Architecture for the input configuration of the input layer.

2.2. Architecture for the hidden layer

The hidden layer is a computational layer where the computation of the between class variance is performed. It is also a layer where the additions, multiplications, and divisions are needed to assist computation. For instance, the hidden layer has fifteen blocks of processing elements. Each processing element is used to find the between class variance σ_B^2 . As shown in Fig. 3 observe that many additions, multiplications, and divisions have to be used in each processing element of the output layer. Also observe closely that some values are calculating more than once in the design. The reason for this repetition in the design is that we have a trade off in parallelism with respect to the amount of hardware we have to use. The justification for

8.2.4

that is, by using more hardware we can accomplish the processing time in a rapid manner by simultaneously calculating the values. When the calculation is done, each hidden layer neuron will forward its value to all the output neurons in the output layer.

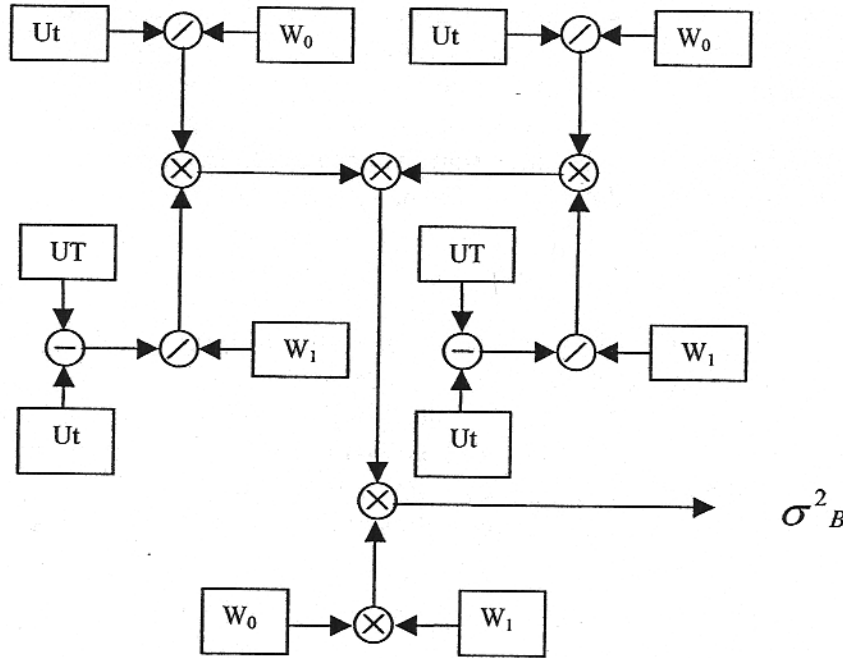


Fig. 3. Architecture for the computation of σ^2_B

2.3. Architecture for the output layer

The output layer is a discrimination layer where each output neuron is connected to itself and fully connected to each other. The output layer will attempt to find the maximum of the values in the hidden layer by performing discrimination using the modified MAX-NET. The architecture of a single neuron in the output layer is shown in Fig. 4. Initially, the value from the hidden layer will be saved in 16 registers. After that, each neuron is updated by

$$\mu_i(t+1) = f[\mu_i(t) - W_{ij} \sum_{j=1}^M \mu_j] \quad (4)$$

The weight is updated after completing each iteration by,

$$W_{ij}(t+1) = \frac{1}{\text{number of neuron firing}} \quad (5)$$

Finally, after the maximum input is found, which is when one neuron is active, the σ^2_B will be compared to CLF. If the value is less than σ^2_B , the process is completed. Otherwise, each output neuron will output 16-bit mask to all neurons in the input layer where the each input

register is enabled or disabled. The mask will be the output in such a way that only i neuron down to the index 0 neuron will be 'ON' and not otherwise.

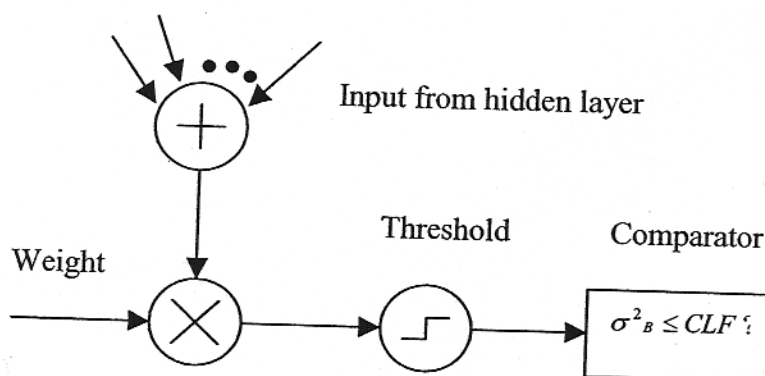


Fig. 4. A single modified MAXNET neuron's architecture

To verify the results of APT, the APT was applied onto a large set of endoscopic images. The images of gastrointestinal tract were captured by an endoscopy system consisting of a miniature CCD camera having a resolution of 200,000 pixels. However, the size of the processed image for the construction of the hardware was restricted to 32×32 pixels in the design. When a colour image is captured, it is transformed to a grey level image consisting of 16 greyscale. The APT technique is independent of the absolute grey level of the pixels contained in the darkest region and hence gives much better results than a thresholding technique that solely depends on the hills and valleys in the image histogram.

The architecture is designed to utilize the concept of neighborhood data sharing. The circuit is simulated in VHDL using Altera Quartus II for implementation in Apex II FPGA. The total number of logic cells utilized for the parallel implementation of APT is 42000. It has been observed that, though the VLSI architecture needs more computing modules, the recurrent neural network converges in two or three iterations by performing repeated recycling of the APT process to achieve optimum threshold, making it suitable for real-time applications.

3 Conclusion

In this paper, we have presented Adaptive Progressive Thresholding (APT) for the real-time segmentation of endoscopic images by extending Otsu's approach and its parallel implementation in hardware. After each iteration, the proposed algorithm segments the object of lowest intensity from a given image. This process continues until the CLF become less than a specified value. Simulation result shows that the technique works flawlessly. A parallel

implementation of the APT was developed, and on this implementation, although the hardware required is more, the convergence time for the architecture is observed to be two or three iterations.

References

- [1] K.V. Asari, C.Eswaran, "Systolic array implementation of artificial neural networks," *Journal of Microprocessor and Microsystems*, vol. 18, pp. 581-488, Oct. 1994.
- [2] K.V. Asari, T. Srikanthan, S. Kumar, D. Radhakrishnan, "A pipeline architecture for image segmentation by adaptive progressive thresholding," *Journal of Microprocessor and Microsystems*, vol. 23, pp. 493-499, Dec. 1999.
- [3] K.V. Asari, "A fast and accurate segmentation technique for the extraction of gastrointestinal lumen from endoscopic images," *Journal of Medical Engineering & Physics*, vol. 22, pp. 89-96, July 2000.
- [4] B. Slatkin, J. Burdick, W. Grundfest, "The development of a robotic endoscope," *Proc. IEEE International Conference on Robotics and Automation*, pp. 162-171, 1995.
- [5] R. Dekleck, J. Cornelis, M. Bister, "Segmentation of medical images," *Journal of Image Vision Computing*, vol. 11, pp. 486-503, 1993.
- [6] K.V. Asari, S. Kumar, D. Radharakrishnan, "Technique of distortion correction of endoscopic images by using a polynomial expansion," *Journal of Medical & Biological Engineering & Computing*, vol. 37, pp. 8-12, Jan. 1999.
- [7] K. V. Asari, S. Kumar, and D. Radhakrishnan, "A new approach for nonlinear distortion correction in endoscopic images based on least squares estimation," *IEEE Trans on Medical Imaging*, vol. 18, no. 4, pp. 345-354, April 1999.
- [8] C. A. Glasbey, "An analysis of histogram-based thresholding algorithm," *CVGIP: Graph Mod Image Processing*, vol. 55, pp. 532-537, 1993.
- [9] H. S. Wu, J. Gil, J. Barba, "Optimal cell segmentation of cell images," *IEE Proc Vision and Image Signal Processing*, vol. 145, pp. 50-56, 1998.
- [10] M. Cheriat, J. N. Said, C. Y. Sue, "A recursive thresholding technique for image segmentation," *IEEE Trans Image Processing*, vol. 7, pp. 918-21, 1998.
- [11] N. Otsu, "A threshold selection method from grey level hostrogram," *IEEE Trans Systems Man and Cybernetics*, vol. 8, pp. 62-66, 1978.

- [12] D. M. Tsai, Y. H. Chen, "A fast histogram-clustering approach for multi-level thresholding," *Pattern Recognition Letter*, vol. 13, pp. 245-52, 1992.