A SYSTOLIC ARCHITECTURE FOR IMAGE SEGMENTATION BY
ADAPTIVE PROGRESSIVE THRESHOLDING

K V Asari¹, T Srikanthan¹, S Kumar², and D Radhakrishnan¹

¹Nanyang Technological University, ²National University of Singapore, Singapore

ABSTRACT

A special purpose VLSI architecture for the real time segmentation of endoscopic images is proposed in this paper. The architecture is based on a systolic implementation of a new algorithm named adaptive progressive thresholding (APT) that segments the darkest area of an endoscopic image representing the gastrointestinal lumen. This segmentation process is an extension of the Otsu’s method. The progressive approach of thresholding optimizes the threshold for an endoscopic image using an iterative procedure. A condition to converge the iterative procedure is suggested to make the segmentation automatic and adaptive. The APT algorithm is mapped onto a modified linear systolic array of simple processing elements with the elements of a particular segment communicating with its neighbors. The APT architecture is partitioned based on various sequential functions involved in the segmentation process and these functional modules are organized in a pipelined fashion according to their hardware feasibility. Currently, a prototype of the VLSI architecture for an image size of 256×256 is being designed and built. The functional simulation results obtained in the APT architecture are encouraging.

INTRODUCTION

Video endoscopy has become an important form of therapeutic procedure during recent times. Typical endoscope consists of a miniature CCD camera that captures the images of gastrointestinal (GI) tract. The automated system proposed in Kumar et al (1) is a vision-guided microrobotic device that necessitates high-speed extraction of various quantitative parameters from the images for the purpose of real-time navigation and path planning. The lumen region of GI images form the preliminary basis of the features used for navigation and automatic guidance. Since the endoscope uses several light sources at its tip and the illuminating distances of these sources are limited, the colon 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.

Several researchers have presented various methods of extracting the lumen region from the endoscopic images. Khan and Gillies (2) used the property of lumen to be the darkest region in the image to segment it from the rest of the image. Segmentation of the region of interest from a given medical image using edge information has been described by DeKlerck et al (3). Glassey (4) has presented a comprehensive analysis of histogram based thresholding algorithms. Cheriat et al (5) have proposed a recursive thresholding technique for image segmentation. The techniques mentioned above do not yield desirable results for the real time segmentation of endoscopic images due to their unstructured environment and unpredictable variations. This leads to the development of a new segmentation technique for lumen extraction. In this paper we propose an adaptive progressive thresholding technique for lumen extraction from a gray level endoscopic image. Otsu (6) has presented a comprehensive statistical technique to find the best threshold for an image. But his method gives good results only with those objects that have sharp contrast between background and foreground. It was also seen that for the endoscopic images a single threshold obtained by Otsu’s method was not sufficient. Hence we use a progressive thresholding approach that is similar to (5). A Cumulative Limiting Factor is also defined for automatic optimization of the threshold to make the process adaptive.

It was observed that the implementation of APT on a PC platform was undesirable due to the large processing time, making online guidance of the microrobot infeasible. Thus, to reduce the overall time taken in the procedure a VLSI architecture is proposed. In addition, the VLSI based system will also facilitate the use of a dedicated hardware that could be mounted along with the endoscopy unit. The modified systolic array is found to be more appropriate as it permits the use of effective pipelining. Many systolic implementation schemes for image processing have been presented. A systolic algorithm and architecture for image thinning has been presented in Ranganathan and Doreswamy (7). The thinning algorithm is mapped onto a linear systolic architecture in which the computation is fully overlapped with I/O to perform online thinning. The implementation of a systolic neural network image processing architecture to perform motion analysis and range estimation on a real time video signal is explained in Cavaino et al (8). In this paper we propose a systolic architecture for APT. The systolic algorithm partitions the entire segmentation process into several functional modules. Our special purpose VLSI architecture uses an efficient encoding strategy that reduces the processing time as well as the communication time between the processors. Each processing cell in the systolic array is simple and requires minimal control logic and the entire architecture can be realized in a single VLSI chip.
ADAPTIVE PROGRESSIVE THRESHOLDING

The method suggested by Otsu is based on discriminant analysis that partitions the image into two classes \( G_0 \) and \( G_1 \) at gray level \( t \) such that \( G_0 = \{0, 1, 2, ..., t\} \) and \( G_1 = \{t+1, t+2, ..., L-1\} \), where \( L \) is the total number of gray levels in the image. Let \( \sigma_B^2 \) and \( \sigma_T^2 \) be between-class variance and total variance respectively. The optimum threshold \( t' \) can be obtained by maximizing the between-class variance. Hence,

\[
t' = \text{Arg Max}_{t \in [0, L]} \{ \eta \}
\]

where,

\[
\eta = \frac{\sigma_B^2}{\sigma_T^2}
\]

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

\[
\sigma_T^2 = \sum_{i=0}^{L-1} (i - \mu_T)^2 p_i
\]

where, \( w_0 \) and \( w_1 \) denote the fraction of pixels lying in \( G_0 \) and \( G_1 \), respectively, and can be represented by,

\[
w_0 = \frac{1}{N} \sum_{i=0}^{L-1} p_i, \quad w_1 = 1 - w_0
\]

\[
p_i = \frac{n_i}{N}, \quad N = \sum_{i=0}^{L-1} n_i
\]

and \( n_i \) is the number of pixels on \( i \)-th gray level and \( N \) is the total number of pixels in the image. The class means for \( G_0 \) and \( G_1 \) respectively and are calculated by,

\[
\mu_0 = \frac{\sum_{i=0}^{L-1} i p_i}{w_0}, \quad \mu_1 = \frac{\mu_T - \mu_0}{1 - w_0}
\]

where,

\[
\mu_T = \frac{1}{L} \sum_{i=0}^{L-1} i p_i, \quad \mu_T = \frac{\sum_{i=0}^{L-1} p_i}{L}
\]

This procedure requires iterative computation to find the best threshold for a given image. Otsu's method of finding gray level images is efficient for separating an image into two classes where two types of fairly distinct classes exist in the image. This property of Otsu's method makes it suitable for being extended so as to provide acceptable thresholding limits for endoscopic images. As the Otsu's approach tries to maximize the between-class variance for the optimized threshold, the threshold image does not correspond to the darkest areas in the endoscopic image. To obtain the darkest region, Otsu's technique is applied recursively on the previously thresholded image. The APT algorithm formulation can be described as below:

Let the image be constituted of \( L \) gray levels varying as the set \( G = \{0, 1, 2, ..., L-1\} \) and is of size \( M \times N \) pixels. Let, \((x, y)\) be the spatial location of a pixel in the image and the corresponding gray level is given by \( g(x, y) \). Therefore the image can be presented as \( g : M \times N \rightarrow G \). It has been noted that due to non-uniform illumination, the endoscopic images carry locally distributed noise which can be eliminated by passing the image through a smoothing filter \( W \) of size \( w \times w \). This facilitates more accurate thresholding results. Thus after smoothing the new intensity at the pixel location \((x, y)\) can be given by,

\[
g'(x, y) = \frac{1}{w^2} \sum_{(x+k, y+l) \in W} g(x+k, y+l)
\]

\( \forall (x, y) \in M \times N \)

The filtered image \( g' \) is subjected to Otsu's method. The separability factor is chosen to be \( \eta \) as defined in Eqn. (1). The gray level \( t' \) which gives maximum value of \( \eta \) is taken as the threshold. The image is further divided into two classes using \( t' \) and a new image \( g'^{(0)} \) is generated so that all the pixels in the original image having gray level more than \( t' \) are excluded from \( g'^{(0)} \). Thus the pixels contained in \( g'^{(0)} \) will have a range \( G'^{(0)} \) given by \( \{0, 1, 2, ..., t'\} \). This procedure is used recursively and a separability factor called Cumulative Limiting Factor (CLF) is used to find appropriate threshold after each iteration. Similar to \( \eta \), CLF is defined as

\[
\text{CLF}(\lambda) = \frac{\sigma_B^2(\lambda)}{\sigma_T^2(\lambda)} \quad \text{for} \; \lambda \geq 1
\]

Here \( \sigma_B^2(\lambda) \) is calculated as given by Eqn. (3) by using \( w_0, w_1, \mu_0 \) and \( \mu_1 \) from the progressive image \( g'^{(0)} \). In this manner, appropriate threshold \( t'(\lambda) \) is obtained by maximizing the value of \( \text{CLF}(\lambda) \) for the image \( g'^{(0)} \). The iterative procedure is stopped whenever

\[
\text{CLF}(\lambda) < \alpha \frac{\mu_T}{\sigma_T^2}
\]

where \( \alpha \) is a constant known as limiting parameter which can be trained using a number of images taken from a particular endoscopic camera. This takes into account the light intensity and distribution of the light sources mounted on the endoscope. The condition given in RHS is chosen by experimenting with a large number of endoscopic images. This type of selection makes the terminating condition image dependent and yields good results for endoscopic image segmentation. The final \( t' \) is taken as the optimum threshold level for the given image. The various computational steps involved in the APT algorithm are shown in Fig. 1.

SYSTOLIC ARCHITECTURE FOR APT

The systolic architecture consists of a pipeline of cells, each cell capable of carrying out some simple computation. The I/O occurs only at the boundary cells. The systolic architecture for APT based on a linear array model with only nearest neighbor interconnections is developed for an 8-bit gray level image of size 256×256. Some variations in the computational steps of the APT algorithm are adopted to make the hardware implementation feasible. Since \( N \) is a fixed number the value of \( w_k \) is computed as the sum of \( n_i \) for \( i = 0 \) to \( t \) and \( w_k = (N-w_0) \). The class mean \( \mu_k \) and total mean \( \mu_T \) are also computed as sum of \( n_i \) for \( i = 0 \) to \( t \) and \( i = 0 \) to \( (L-1) \) respectively.
Architecture for Histogram and Intensity Area Computation

The architecture developed for the computation of cumulative histogram (CH) and cumulative intensity area (CIA) is shown in Fig. 2. The first stage in the architecture is the intensity histogram computation module. The 64Kbyte image data is stored in four 16Kbyte RAMs \( R_1, R_2, R_3, \) and \( R_4 \) and the data reading process is pipelined. The read signals of the RAMs are multiplexed in such a way that RAM reading time overlaps after every single computation period of the intensity histogram (IH) counter array which holds \( n_i \) for all \( i \) and intensity area (IA) accumulator array which holds \( i_n \) for all \( i \). Further a decoder receives data from RAMs \( R_1, R_2, R_3, \) and \( R_4 \) in a sequential order at every T/4 interval, where T is the RAM read time. During the first T/4 duration the decoder decodes \( I_0 - I_{15} \). Then it decodes \( I_{16} - I_{31} \) during the second T/4 duration, \( I_{32} - I_{47} \) during the third T/4 duration and \( I_{48} - I_{63} \) during the fourth T/4 duration. The 256 outputs of the decoder are connected to the IH accumulators and a counter which increments its value according to the intensity level of the input pixel. After the entire image data from RAMs is read in, the histogram information will be available in 256 counters. The intensity area computation is then performed by transferring the IH contents to the multipliers. The index of the IH data is stored in separate registers for the IA computation and the products are stored in the IA accumulators.

The second stage of the pipeline computes IH and IA accumulation. For this purpose the IH and IA registers are circularly coupled so that the content of each register can be transferred to its top nearest neighbor after every circular shift. The IH accumulation (IHA) and IA accumulation (IAA) are performed by successive addition and accumulating the sums in the ACC registers in each circular shift operation. The IHA and IAA data are pushed into two register arrays having 256 registers each. After 256 circular shift operations in IH and IA registers, the contents of the \( t \)th register in the IHA and IAA register arrays will be,

\[
\text{IHA}(t) = \sum_{i=0}^{t} n_i, \quad \text{IAA}(t) = \sum_{i=0}^{t} i n_i
\]

Architecture for Between-Class Variance Computation

The second layer of the APT architecture developed for the computation of between-class variance is shown in Fig. 3. The IHA and IAA register arrays are divided into 16 blocks with 16 registers in each block. These register blocks function independently as 16 different arrays. Reg1 and Reg2 hold the top contents of the working IAA and IHA respectively for computing the value of \( \alpha i \), during each progressive computation step. Each block has an address counter to hold the index of the current register transferring the data. The \( i \)th block of IHA register array is circularly shifted and the values of \( w_0, w_i \), and \( w_t \) are computed. Similarly from IAA, \( w_0 \) and \( w_i \) are computed where \( w_i \) represents the sum of \( \text{(i.n)} \) for \( i = 0 \) to \( t \) and \( w_i = (\mu_i - w_0) \). Then the class means \( \mu_0 \) and \( \mu_1 \), representing \( G_0 \) and \( G_1 \) classes are computed using divider circuits. The between class variance \( \sigma_b^2 \) is computed by successive multiplication. The \( \sigma_b^2 \) values are then passed to a maximization circuit to obtain \( \sigma_b^2 \). Similarly 16 parallel blocks perform their computations simultaneously in a pipelined fashion for the computation of \( \sigma_b^2 \) for \( i = 1 \) to 16.

Optimum Threshold Computation Architecture

The register arrays holding the contents of 16 different values of \( \sigma_b^2 \) for \( i = 1, 2, ..., 16 \) and the corresponding \( t' \) are then circularly shifted to enable the computation of the maximum value \( \sigma_b^2 \) as shown in Fig. 4. If the maximum value obtained is less than \( \mu_{01} \), the progressive threshold operation is activated and the threshold \( t' \) obtained is used as the top value of the IHA and IAA. The contents of these top registers are then transferred to Reg1 and Reg2 and all registers above this index in IHA and IAA are reset to '0'. Furthermore, a new \( t' \) for the maximum \( \sigma_b^2 \) of the new IHA and IAA is computed as above. If the new \( \sigma_b^2 \) does not satisfy the stipulated condition, this process is repeated till the optimum threshold is obtained.

RESULTS AND DISCUSSION

To verify the results of segmentation, the proposed algorithm was applied onto a large set of endoscopic images. The images of gastrointestinal tract were captured using an endoscope consisting of a miniature CCD camera having a resolution of 200,000 pixels. The typical size of the processed image was restricted to 256 x 256 pixels. A gray level image consisting of 256
Fig. 2 Architecture for computation of cumulative histogram and intensity area

Fig. 3 Architecture for computation of between-class variances

Fig. 4 Architecture for computation of $\sigma_{B_{\text{max}}}^2$ and final threshold
levels was generated from the original color image and a filter of size $3 \times 3$ was used to smoothen the image.

![Original image](image1.png) ![Thresholderd image](image2.png)

**Fig. 5** Segmentation of a typical endoscopic image

The APT technique is independent of the absolute gray 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 optimum value of limiting parameter $\alpha$ was chosen using a training set of images and was found to be 8.3 in our application. The optimum threshold was found to be 31. A typical endoscopic image and the thresholded image are shown in Fig. 5a and 5b respectively. It can be noted that the APT gives a fairly accurate result in segmenting the lumen region from the rest of the image. The outer areas of the segmented region may deviate slightly from the exact boundary due to the smoothening of the transitional areas near the boundaries but it does not significantly affect the segmentation accuracy.

In the APT architecture, the entire process of segmentation was partitioned into different stages according to the individual functions. Each functional module transferred data to its nearest neighbor after completing its computation. The time required to finish the entire operation of generating IAA and IHA from the original image, to obtain the first optimum threshold after the histogram and intensity area computations, was only 288 circular shift cycles. The main fraction of the total time elapsed was in histogram computation. During this time 10K memory read operations were performed. But due to the split memory strategy adopted for the histogram computation architecture, the memory read time is reduced to one-fourth. Further the progressive computations were initiated and time required for these reduce with iterations. The total time taken for segmentation, while implementing APT on a PC platform (Pentium, 200MHz) for the image shown in Fig. 5a was found to be 40ms. As the segmentation is a preliminary step used before an extended image analysis procedure, the time taken by APT precludes real time navigation strategy. But the simulation results of the proposed hardware shows that the overall time taken was about 0.16ms for the same image. This comparison was carried out for a large number of images and it was observed that the overall time taken reduces by approximately a factor of 1/256. However, it can be seen that in the present implementation systolic arrays are control driven and therefore require correct timing between processors. This timing is more complex than a global clock on an SIMD machine because different cells compute different operations.

**CONCLUSION**

In this paper we have proposed a new adaptive technique for the real time segmentation of endoscopic images by extending Otsu’s approach and an efficient architecture to implement it in hardware. After each iteration, the proposed algorithm segments the object of lowest intensity from a given image. This process continues until the CLF becomes less than a specified value. Simulation results obtained on endoscopic images indicate the effectiveness of the technique. A systolic architecture for the implementation of the APT technique was developed. In architecture, the total time for the pipeline to complete the segmentation process was found to be about 1/256 of that required by the conventional technique. The functional simulation results obtained for an image of size 256x256 proves that compute bound problems can be implemented efficiently in a systolic architecture. Further work is in progress to convert this into a reconfigurable structure so that the different sequential modules in the system can be partitioned so that an area efficient architecture can be built.

**REFERENCES**


