A NEW PARALLEL THINNING ALGORITHM FOR GRAY SCALE IMAGES

Samira S. Mersa1[1] and Ahmed M. Darwish[2]

[1]Mathematics and Scientific Computing Department
Suez Canal University
Ismailia, EGYPT
smersal@frcu.eun.eg

[2]Computer Engineering Department
Cairo University
Giza, 12613 EGYPT
darwish@frcu.eun.eg

ABSTRACT

In this paper we introduce a parallel algorithm for thinning gray scale images. The algorithm is based on repeatedly conditionally eroding the gray level objects in the image until a one pixel thick pattern is obtained along the center of the high intensity region. Erosion conditions are devised to assure preserving connectivity. To properly handle objects with hollows, gray level gradient is used in the preprocessing phase to detect significant hollows. This allows processing of realistic nontrivial gray level objects. Results of applying the algorithm on a variety of images will be shown.

Key words: Thinning, Skeleton, Gray level images, Erosion, Parallel algorithms.

1. INTRODUCTION

Thinning and skeletonization have numerous applications in image analysis and computer vision. For several of these applications significant amount of information is lost during the process of binarization. Applying thinning directly to gray scale images is motivated by the desire of directly processing images with gray levels distributed over a range of intensity values. This will avoid shape distortions that may irremediably affect the presence of features in the binary image generated even if an optimal thresholding algorithm is used to produce the binary image.

The gray skeleton is a connected subset of a gray scale pattern which consists of a network of lines and arcs centrally placed along local higher intensity regions. Unfortunately, there is no one single agreed upon definition for gray skeletons [1], [2], [8]. Skeletons are classically associated as a medial axis representation that is regenerative (i.e. could be used to generate the object back exactly). Skeletons are not easily digitizable. It is not possible to have a representation that is a medial axis, preserves connectivity, preserves homotopy and exists on the square digital grid. One of these four restrictions has to be relaxed. This has opened the way to several approximations known as thinning.

We can group most of the published algorithms under one of two approaches. The first approach considers the image as a continuous surface in the 3D Euclidean space and use the first and second partial derivatives of this surface to assign the proper topographical label to each pixel [3], [15]. The second approach is based on the repeated application of a removal process that erodes the gray scale pattern until only one pixel thick subset is obtained in the center of the high intensity region. The algorithm proposed in this paper is a parallel thinning algorithm that preserves connectivity and belongs to the latter family. We have devised a set of conditions that guarantee that the resulting thinned version is connected and as close as possible to the medial axis. The algorithm has been tested on a variety of images from different applications and produced satisfactory results that proved to be useful for compression and recognition applications which will be reported elsewhere.

Aside from this introductory section, the rest of the paper is composed of 4 more sections. In section 2 we give a quick review of previously published work in the field. Details of the suggested algorithm are presented in section 3. Section 4 shows results of applying the algorithm to images from different applications. We finally conclude in section 5 with observations and recommendations for future work.

2. LITERATURE SURVEY

Thinning and skeletonization of binary images have been studied extensively since the early sixties. In the case of gray scale images, the literature does not provide a general agreement on the requirements for a gray skeleton in order to constitute a meaningful representation. The result of an algorithm is dependent on the definition of connectivity. Different definitions of local gray scale connectivity were presented. One definition, [14], states that the neighborhood of a pixel, p, is connected if the connective strength of any pair in the neighborhood of p is not less than p. A slightly different definition given in [4] considers the neighborhood of a pixel, p, to be connected if the connective strength of any pair in the neighborhood is not less than both values of the pair even if these values are less than the value of p. While in [1] the condition was imposed such that it is not less than both the pair and p. Another approach taken by [7] is to threshold the neighbors of p, then use the binary connectivity definition.

This is not intended as a survey paper, but for the convenience of the reader and the ease of comparison, in the following we will present some of the gray scale skeletonization and thinning algorithms documented in the literature almost in historical order.

Levi and Montanari [8] presented a gray weighted skeleton based on the concept of gray weighted distance. The gray weighted distance is proportional to the sum of the gray levels along the path. The skeleton is the set of all points which do not belong to any minimal gray weighted path from any other point to the background. The skeleton resulting from this algorithm does not guarantee the connectivity. Dyer and Rosenfeld [4] presented a parallel algorithm. Their definition of connectivity does not provide global connectivity of the skeleton, moreover the skeleton does not lie along the high gray values but it is positioned in a central place determined by the boundary of the image. Peleg and Rosenfeld in [13] proposed a Min-Max medial axis transformation. Salari and Siy, [14], presented a two phase sequential algorithm. In the first phase they computed the contextual gray distance transformation (CGDT). In the second phase they removed the boundary pixels which satisfied a set of conditions. The algorithm required the input image to be segmented into zero and non-zero pixels. The resulting skeleton is one pixel wide positioned in the ridge areas.

Abe et al., [1], point out the problems and resulting defects with the Salari and Siy algorithm, [14], due to its pure sequential processing nature and presented a combined sequential and parallel algorithm. Their algorithm is considered an extension of the Hilditch algorithm, [5]. Maragos and Ziff, [10], compute the gray skeleton by summing the skeletons of the binary images that result from thresholding the gray scale image at each value of the gray scale. Low, [9], defined gray skeleton in terms of gray mathematical morphology. Pal and Ghosh, [12], used three functions h(p), v(p) and f(p) which represent the horizontal, vertical membership functions and the degree of brightness respectively. The three functions are combined in different ways to define g(p) which denotes if the pixel belongs to the core line in the objects.

Arcelli and Ramella, [2], presented a parallel thinning algorithm. They put two implementation based on the operations R1 and R2. R1 denote the sequence of four parallel operations, each operation remove pixels that satisfy certain conditions. Thinning is accomplished by repeatedly applying the sequence of the four operations to remove north, east, south and west border points respectively. The process terminates when no pixel is removed during a whole sequence. While R2 is a parallel operation that removes pixels which satisfy another set of conditions. The thinning is accomplished by repeatedly applying R2 until no further pixel is removed. They observed that the two operations R1 and R2 produce largely similar skeletons. Once difference is that the algorithm based on operation R1 is more prone to the creation of skeleton branches than the algorithm based on R2.

Skeletonization algorithms based on a topographic approach were presented in [3] and [12]. These algorithms are aimed at avoiding object distortion, reducing deformation on junctions of a skeleton.

3. ALGORITHM

In this section we first introduce some basic definitions, then we detail the three main steps of the algorithm and finally we present algorithm summary and analysis of its computational complexity.

3.1 Definitions

The 8 neighbors of a pixel, p, will be labeled as p_0, p_1, ..., p_7 starting on the north and going clock wise until the north west.

Background is taken as 0 and any other value [1–255] is considered to be a part of the object.

A hollow is a connected flat region, all of its pixels have gray level value k, from which any path to pixels not in the region necessarily includes pixels have gray value greater than k.

Weight number of a pixel, p, is defined as

$$\mathrm{WN(p)} = \sum_{k=0}^{7} p_k * 2^k$$

For 3×3 windows the weight numbers are between [0,255]. The set of all weight numbers is WS = {0, 1, 2, .., 255}.

A border point is a point that has at least one background point in its 4-neighbors.

Connectivity number of a pixel, p, is given by

$$CN(p) = \sum_{k=0,2,4,6} \bar{p}_k \wedge (p_{k+1} \vee p_{k+2})$$

where $\bar{p}_k = 1 - p_k$

Ridges are composed of pixels, p, where CN(p) > 1.

Strength of a Path is the minimum value of the pixels belonging to the path including the start and end points of the path.

A 4-connected (8-connected) strength between a pair of pixels is the maximal strength of the 4-connected (8-connected) paths between those pixels.

3.2 Preprocessing

3.1.1 Image smoothing

This step is used to remove non-significant hollows. It is accomplished by a min-max procedure. Median filters could also be used.

3.1.2 Hollow detection

We use the gradient operators of Sobel or Prewitt [6] to compute the partial derivatives f_x and f_y in the x and y directions respectively. The gradient is calculated as $g(x,y) = |f_x + f_y|$.

The pixel p(x,y) is declared as edge point if $g(x,y)$ exceeds a threshold value t which is selected so that less than 5 percent of the pixels are declared as edges. Each pixel detected as an edge pixel its value is subtracted from all its 4-neighbors. The neighbor pixel which result in the maximum absolute difference is changed to zero if its gray value less than p(x,y) otherwise p(x,y) itself is changed to zero. This step is necessary to introduce a background pixel inside the hollow to start eroding from the inside out.

3.3 Erosion (Border Removal)

The erosion operation is an iterative procedures that removes certain border pixels, i.e. change their gray level value to zero. Rules imposed on the removal or erosion operation must guarantee that it neither destroys connectivity nor reduces the gray scale connective strength. In the proposed algorithm, connectedness is accomplished by a method similar to the method used in the binary algorithm reported in [16].

The gray connectivity strength is preserved by retaining ridge pixels. As defined above, identification of ridge pixels is based on the value of CN. CN is computed based on a binary version of the 3×3 neighborhood of p, such that p_i is set to zero if $p_i < p$ otherwise set p_i to one for all i = 0, 1, ..., 7.

To assure that the resulting skeleton or thinned pattern is near the center of the ridges as much as possible, removing of all qualified pixels is not done in parallel at the same iteration. Instead we introduce two types of removal; unrestricted and restricted. We apply them on alternate iterations.

3.3.1 Unrestricted removal

In this operation all border pixels which are not ridge pixels and are simple pixels are removed.

To check whether a pixel is simple or not WN(p) is computed. Another temporary binarization of the neighborhood of p takes place such that p_i is set to 1 if it belongs to the object otherwise it is set to 0. The following array, [16], is used to check for simple pixels as stated below.

```
WNset[0..255] =
(
0,0,0,2,0,6,1,8,0,0,   0,0,1,5,1,5,0,0,0,0,   1,5,1,5,0,0,0,0,1,5,
1,5,0,0,0,0,0,0,0,0,   0,0,0,0,0,0,0,0,1,0,
                       0,0,1,5,1,5,1,0,0,0,
1,5,1,5,0,2,0,2,0,2,   0,2,0,0,0,0,0,6,0,6,   1,3,0,4,1,0,1,0,1,3,
0,4,1,0,1,0,1,3,0,4,   0,4,0,4,0,0,0,0,0,7,   0,7,1,3,0,4,1,0,1,0,
1,3,0,4,1,0,1,0,0,2,   0,2,0,2,0,2,0,0,0,0,   0,6,0,6,0,0,0,0,0,6,
0,6,0,0,0,0,0,6,0,6,   0,0,0,0,0,0,0,0,0,0,   0,0,0,0,0,0,0,0,0,0,
0,6,0,6,0,0,0,0,0,6,   0,6,1,1,0,2,0,2,0,2,   0,0,0,0,0,6,0,6,1,1,
0,2,1,0,1,0,1,1,0,2,   1,0,1,0,1,1,0,2,0,2,   0,2,0,0,0,0,0,6,0,6,
1,1,0,2,1,0,1,0,1,1,   0,2,1,0,1,0
);
```

```
p is a simple pixel if
WNset[WN(p)]=1
or (WNset[WN(p)]=2     and WNset[WN(0)]=0)
or (WNset[WN(p)]=3     and WNset[WN(6)]=0)
or (WNset[WN(p)]=4     and WNset[WN(0)]=0
                       and WNset[WN(6)]=0)
or (WNset[WN(p)]=5     and WNset[WN(2)]=0)
or (WNset[WN(p)]=6     and WNset[WN(0)]=0
                       and WNset[WN(2)]=0)
or (WNset[WN(p)]=7     and WNset[WN(0)]=0
   and WNset[WN(2)]=0      and WNset[WN(6)]=0)
or (WNset[WN(p)]=8     and WNset[WN(1)]=0).
```

where WN(0), WN(2), WN(6) and WN(1) are the weight numbers of the north, east, west and northeast neighbors of the pixel p.

3.3.2 Restricted removal

The removal operation in this step is broken into several substeps where removal in each substep is restricted to the border pixels which have gray values not greater than a specific value. Assuming there is some correlation between removed pixels in iteration i and iteration i-1, we remove pixels in this case based on the gray level values of the pixels removed in the previous iteration (i-1). The gray values of the pixels removed by the previous unrestricted removal iteration are sorted in ascending order g_1, g_2, ..., g_n. We then repeatedly remove all border pixels that satisfy the conditions in 3.3.1 and have gray value not greater than g_1, then remove all pixels having gray value not greater than g_2 and so on until all the border pixels having gray value not greater than g_n are removed.

3.4. Algorithm Summary

1. Perform image smoothing.
2. Perform hollow detection and introduce a background pixel in each hollow.
3. For every border pixel, p, in the image perform the following (simultaneously if architecture allows):
   Compute the connectivity number CN(p) based on binarization mentioned above in section 3.3.
   If CN(p) > 1
   then
      it is a ridge pixel and can not be removed,
   else
      Compute WN(p)
      If it is a simple pixel (see test in sec 3.3.1),
      then
         remove it, i.e. set its value to zero and keep a record of its gray level,
      else
         keep it as is.
4. Rank the gray level values of the removed pixels in step 3, g_1, g_2, ..., g_n.
5. Repeat n times (i=1, ... n):
   Apply step 3 for all border pixels of gray level $\leq g_i$.
6. If no pixels are removed end, else goto step 3.

3.5. Computational Complexity

The algorithm is a fast one. All operations are simple compare or shift operations. All iterations are of O(N), where N is the number of pixels. The algorithm could be implemented in parallel so that ultimitelay each iteration becomes O(1) for N processors. The number of iterations equals half the thickest object in the image.

4. RESULTS

The algorithm has been tested on a variety of images. In this paper we report results on three test sets. The first test set included document images. The second set consisted of images of 40 mouse chromosomes revealed with Giemsa staining. The images were a set of attached bands with different gray level values. The third set consisted of 24 images of Actinomyces. Figures 1-3 shows sample images.

We have tested the algorithm with different preprocessing options. As expected, it was observed that smoothing reduces the number of branches in the resulting thinned pattern. Also the percentage of pixels used to initialize hollows affected the result very much. For the above test sets it had to be well below 5%.

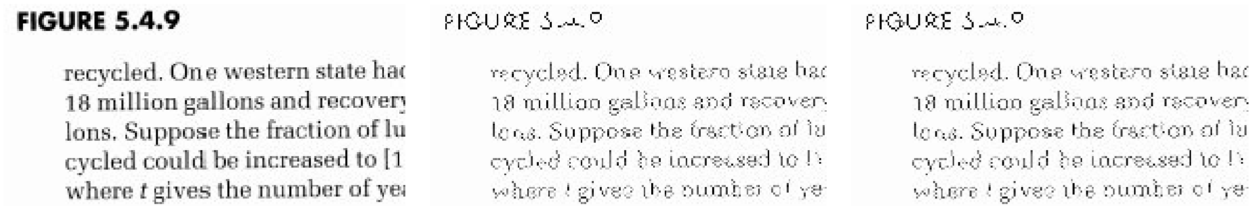

Figure 1. Sample of a part of a document. Left to right, original gray level image, result without the preprocessing steps and results with preprocessing.

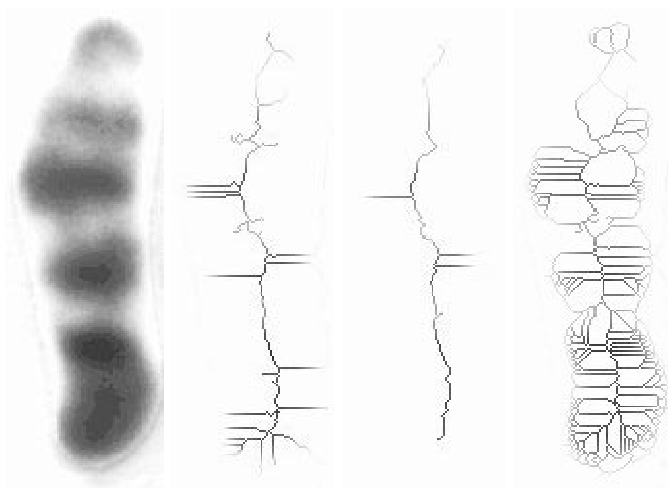

Figure 2. Image of a mouse chromosome. Left to right, original image, results of thinning in three cases; using light smoothing and hollow initialization of 1%, using high smoothing and using light smoothing and increasing hollow initialization to 5%.

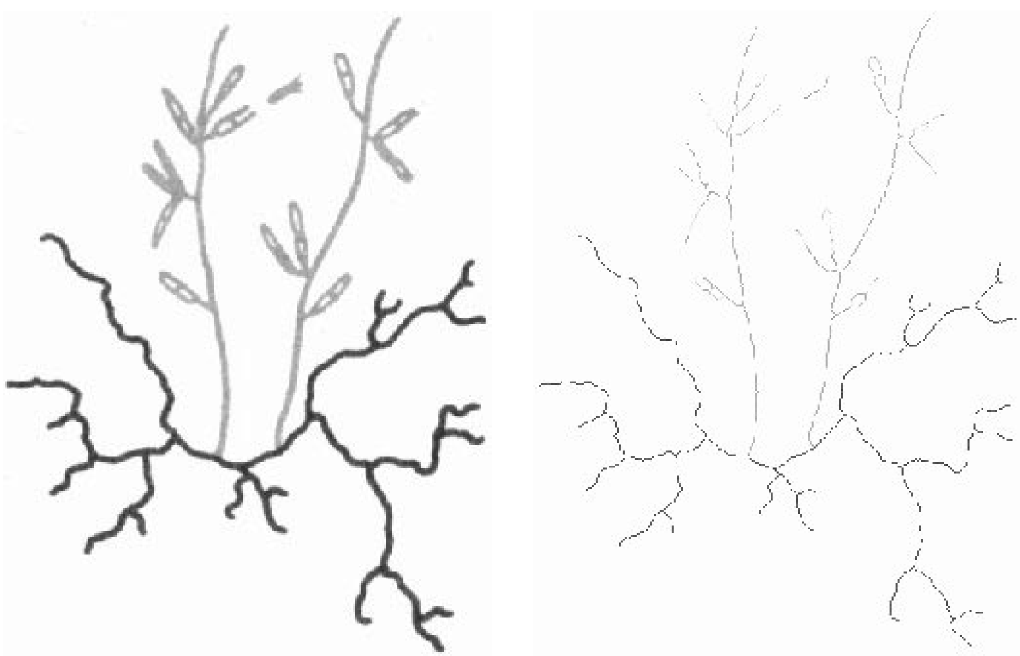

Figure 3. Image of a Planobispora Actinomyces. Left original, right thinned.

In most cases the algorithm produced satisfactory results that proved to be useful for compression and recognition applications. A full system for recognizing and matching chromosomes was based on the algorithm and will be reported elsewhere. For that system a special pruning procedure was devised to remove non-significant branches [11]. The pruning rules are different from those previously mentioned in the literature such as those in [6].

5. CONCLUSION

No doubt that for several applications there are several advantages to be able to thin the gray level objects without converting the image into a binary image. In this paper we have introduced a parallel thinning algorithm for gray scale images. The algorithm is based on eroding objects iteratively by removing certain border pixels without affecting the connectivity. The algorithm was tested on images from different domains and produced satisfactory results.

REFERENCES

[1] Abe, K., Mizutani, F. and Wang, C., "Thinning of gray-scale images with combined sequential and parallel conditions for pixel removal", *IEEE Trans. on Systems Man Cybernetics*, vol. 24, no. 2, pp. 294-299, Feb. 1994.

[2] Arcelli, C. and Ramella, G., "Finding grey-skeletons by iterated pixel removal", *Image and Vision Computing*, vol. 13, no. 3, pp. 159-267, Apr. 1995.

[3] Chen, S. and Shih, F., "Skeletonization for fuzzy degraded character images", *IEEE Trans. on Image Processing*, vol. 5, no. 10, pp. 1481-1485, Oct. 1996.

[4] Dyer, C. R. and Rosenfeld, A., "Thinning algorithms for grayscale pictures", *IEEE Trans. Pattern Anal and Machine Intell.*, vol. 1, no. 1, pp. 88-90, Jan. 1979.

[5] Hilditch, C., "Linear skeletons from square cupboards", *Machine Intelligence IV*, Edinburgh University Press, pp. 403-420, 1969.

[6] Jain, A., *Fundamentals of digital image processing*, Prentice Hall, 1989.

[7] Kundu, M., Chaudhuri, B. and Dutta-Majumder, D., "A parallel greytone thinning algorithm (PGTA)", *Pattern Recognition Letters*, vol. 12, no. 8, pp. 491-494, 1991.

[8] Levi, G. and Montanari, U., "A gray-weighted skeleton", *Information and Control*, vol. 17, pp. 62-91, 1970.

[9] Low, A., *Introductory computer vision and image processing*, 1991.

[10] Maragos, P. and Ziff, R., "Threshold superposition in morphological image analysis systems", *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 12, no. 5, pp. 498-504, May 1990.

[11] Mersal, S., *Mathematical morphology techniques for gray level image processing*, Ph.D. Dissertation, Cairo University, in progress.

[12] Pal, S. and Ghosh, A., "Fuzzy geometry in image analysis", *Fuzzy Sets and Systems*, vol. 48, pp. 23-40, 1992.

[13] Peleg, S. and Rosenfeld, A., " A min max medial axis transformation", *IEEE Trans. PAMI*, vol. PAMI 3, no. 2, pp. 208-210, March 1981.

[14] Salari, E. and Siy, P. "The ridge seeking method for obtaining the skeleton of digital images", *IEEE Trans. on Systems Man and Cybernetics*, vol. SMC-14, no. 3, pp. 524-528, May/June 1984.

[15] Wang, L. and Pavlidis, T., "Direct gray-scale extraction of features for character recognition", *IEEE Trans. on Pattern Anal. And Machine Intell.*, vol. 15, no. 10, pp. 1053-1067, Oct. 1993.

[16] Zhang, Y. and Wang, P., "New parallel thinning methodology", *Intr. Journal of Pattern Recog. & Artificial Intell.*, vol. 8, no. 5, pp. 999-1011, 1994.