Reliable algorithm for thinning number patterns

Anup Lal Yadav[1], Sahil Verma[2], Kamal Kumar Sharma[3]

[1]Student, M. Tech, ESEAR, Ambala

[2]Assistant Professor, Dept. of CSE, E-Max group of Institutions, Ambala

[3]Professor, Dept. of ECE, E-Max group of Institutions, Ambala

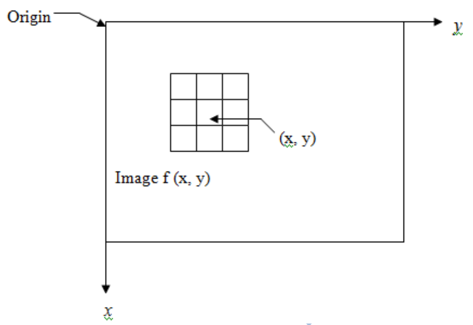
Abstract— **A number of image processing and pattern recognition application demand that are raw digitized binary pattern array be normalized, so that the constituents components of that array are of uniform thickness. The Thinning process reduces such components to a thickness of one pixel or sometimes to a few pixels. The Thinning of binary images has been studied extensively since the early sixties. Sometimes the image features are not directly or obviously associated to any part of the image, but still reflect particular image properties, like the image moments. The selection of image features and corresponding extraction methods is probably the most important step in achieving high performance for an OCR system. At the same time, the image feature and the extraction method also decide the nature and the output of the image-preprocessing step. Some image features and the extraction algorithms work on colour images, while others work on gray level or binary images. Moreover, the format of the extracted features must match the requirements of the classifier. Some features like the graph descriptions and grammar-based descriptions are well suited for structural and syntactic classifiers. The sequential algorithms much of the complexity of the parallel algorithms results from the need to preserve connectedness while using parallel operations in the small local neighborhood. This is a problem particular to nature of parallel algorithm and it has been addressed by using sub iterations or by enlarging the neighborhood to be examined. In either case, intermediate results are actually computed and in some form in order to preserve more global structures.**

Keywords— **Image processing**

.

I. Introduction

A digital image is composed of a finite number of elements, each of which has a particular location and value. These elements are referred to as pixels. The intensity of a monochrome image at any coordinates (x, y) is called the gray level (l) of the image at that point. That is,



I = f (x, y)

A gray level image is composed of pixels in the gray scale defined by the interval [L_{min}, L_{max}]. The common practice is to shift this interval numerically to the interval [0, L-1], where I = 0 is considered black and I = L -1 is considered white on the gray scale[2] An image having only two levels namely black and white (with black denoted by 1 and white denoted by 0) is called a binary image.

Image Structure

An image consists of tiny, equal areas, or picture elements, arranged in regular rows and columns. The position of any picture element, or pixel, is determined on an x-y coordinate system; the origin is at the upper left corner of the image. Each pixel also has a numerical value, called a digital number (DN) that records the intensity of electromagnetic energy measured for the ground resolution cell represented by that pixel. Digital numbers range from zero to some higher number on a gray scale [3]. The position of a pixel P_1 at location (i, j) along with its neighboring pixels is shown in Figure 1.2.

P_9 (i-1, j-1)	P_2 (i-1, j)	P_3 (i-1, j+1)
P_8 (i, j-1)	P_1 (i, j)	P_4 (i, j+1)
P_7 (i+1, j-1)	P_6 (i+1, j)	P_5 (i+1, j+1)

Fig. 1.2: Designation of 9 pixels in 3x3 window

Image Features

Image features are unique characteristics that can represent a specific image. Image features are meaningful, detectable parts of the image[4]. Two types of characteristics are usually referred by image features: -

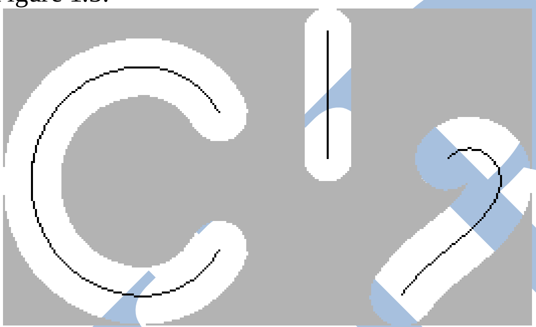
1. A global property of an image, for instance the average gray level of all pixels included in a gray level image.
2. A part of the image with special properties, for example the boundary length of an image.

Sometimes the image features are not directly or obviously associated to any part of the image, but still reflect particular image properties, like the image moments. The selection of image features and corresponding extraction methods is probably the most important step in achieving high performance for an OCR system. At the same time, the image feature and the extraction method also decide the nature and the output of the image-preprocessing step. Some image features and the extraction algorithms work on colour images, while others work on gray level or binary images. Moreover, the format of the extracted features must match the requirements of the classifier. Some features like the graph descriptions and grammar-based descriptions are well suited for structural and syntactic classifiers [5].

Thinning

Thinning is a morphological operation that is used to remove selected foreground pixels from binary images[7]. It can be used for several applications, but normally only applied to binary images, and produces another binary image as output. The Thinning operation is related to the hit-and-miss transform, and so it is helpful to have an understanding of that operator before reading on.

The term ‘skeleton’ has been used in general to denote a representation of a pattern by a collection of thin arcs and curves. Other nomenclatures have been used in different context. For example the term ‘medial axis’ is being used to denote the locus of centers of maximal blocks Some authors also refer to a ‘thinned image’ as a line drawing representation of pattern[8]. In recent years, it appears that the term ‘skeleton’ is used to refer to the result, regardless the shape of the original pattern or the method employed[9]. Thus, Thinning is defined as process of reducing the width of pattern to just a single pixel. This concept is shown in Figure 1.3.



The skeletons of one object may belong to several other different elongated objects as shown in Figure 1.4.

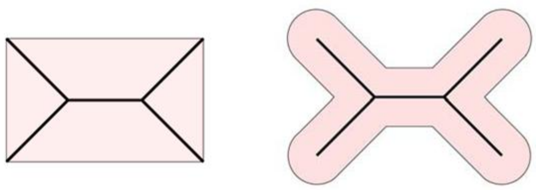


Fig. 1.4: The different objects with the same skeletons

II. How It Works

Like other morphological operators, the behavior of the Thinning operation is determined by a structuring element. The binary structuring elements used for Thinning are of the extended type described under the hit-and-miss transform (*i.e.* they can contain both ones and zeros). The Thinning operation is related to the hit-and-miss transform and can be expressed quite simply in terms of it. The Thinning of an image *I* by a structuring element *J* is:

$$thin(I, J) = I - \text{hit-and-miss}(I, J)$$

In everyday terms, the Thinning operation is calculated by translating the origin of the structuring element to each possible pixel position in the image, and at each such position comparing it with the underlying image pixels. If the foreground and background pixels in the structuring element *exactly match* foreground and background pixels in the image, then the image pixel underneath the origin of the structuring element is set to background (zero). Otherwise it is left unchanged. Note that the structuring element must always have a one or a blank at its origin if it is to have any effect. The choice of structuring element determines under what situations a foreground pixel will be set to background, and hence it determines the application for the Thinning operation. For example, consider the structuring elements as shown in Figure 1.5.

0	0	0
	1	
1	1	1

	0	0
1	1	0
	1	

***Fig. 1.5: Example** structuring elements for Thinning.*

At each iteration, the image is first skeletonized by the left hand structuring element, and then by the right hand one, and then with the remaining six *90°* rotations of the two elements. The process is repeated in cyclic fashion until none of the skeleton produces any further change. As usual, the origin of the structuring element is at the center.

Figure 1.6 shows the result of this Thinning operation on a simple binary image.

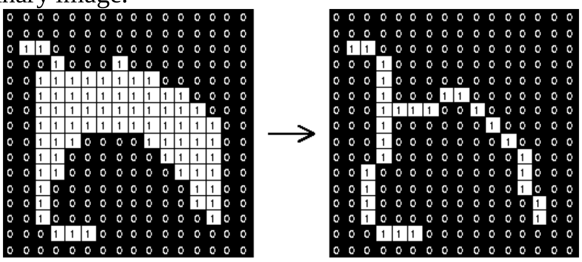


***Fig. 1.6: Thinning** of a simple binary shape, using the above structuring elements.*

We have described the effects of a single pass of a Thinning operation over the image. In fact, the operator is normally applied repeatedly until it causes no further changes to the image (*i.e.* until *convergence*). Alternatively, in some applications, *e.g. pruning*, the operations may only be applied for a limited number of iterations.

Purpose of Thinning

The purpose of Thinning is to reduce the amount of information in image pattern to the minimum needed for recognition. Thinned image helps the extraction of important features such as end points, junction points, and connections from image patterns[10]. In real world there is a need for Thinning of images due to following reasons:

- To reduce the amount of data required to be processed.
- To reduce the time required to be processed.

- Extraction of critical features such as end-points, junction-points, and connection among the components.
- The vectorization algorithms often used in pattern recognition tasks also require one-pixel-wide lines as input.

Shape analysis can be more easily made on line like patterns.

III. Thinning Approaches

Thinning is defined as a procedure to transform a digital pattern, say, a connected component, to a connected skeleton of unit width. Thinning is normally only applied to binary images, and produces another binary image as output. Two major approaches of skeletonizing digital patterns can be categorized into:

- Iterative boundary removal algorithms, and
- Distance transformation algorithms.

Iterative boundary removal algorithms delete pixels **on** the boundary of a pattern repeatedly until only unit pixel-width thinned image remains. These algorithms remove the boundary pixels of a connected component that neither is essential for preserving the connectivity of the pattern nor do they represent any significant geometrical features of the pattern[11]. The process converges when the connected skeleton does not change or vanish even if the iteration process continues.

The Thinning algorithm, iteratively applied to subsequent binary images, consists of successive iterations of two basic steps, applied to the contour points, to select pixels to be deleted. A contour point is any pixel with value 1 having at least one 8-neighbour valued '0'. The algorithm uses masks in order to select pixels to be deleted. The 8 closest neighbors are numbered following a clockwise walk around the pixel *P*, starting at the upper edge as shown in Figure 1.7.

P_8	P_1	P_2
P_7	P	P_3
P_6	P_5	P_4

Fig. 1.7: Neighbors of pixel P

Distance transformation algorithms are not appropriate for general applications since they are not robust, especially for patterns with highly variable stroke directions and thicknesses.

IV. LITERATURE SURVEY

A number of image processing and pattern recognition application demand that are raw digitized binary pattern array be normalized, so that the constituents components of that array are of uniform thickness. The Thinning process reduces such components to a thickness of one pixel or sometimes to a few pixels. The Thinning of binary images has been studied extensively since the early sixties.

During these years, many Thinning algorithms for data compression by Thinning have been devised and applied to a great variety of patterns for different purpose. For example, in the biomedical field, this technique was found to be useful in the early 1960's, when a "shrink" algorithm was applied to count and size the constituent parts of white blood cells in order to identify abnormal cells. Since that time, applications in this area have included analysis of white blood cells and chromosomes, automatic X-ray image analysis and pattern recognition.

This wide range of application shows the usefulness of reducing the patterns to thin line representations, which can be attributed to the need to process a reduced amount of data as well as to the fact that shape analysis can be more easily made on line like patterns.

T. Y. ZHANG et. al [3] in their research paper *"A fast parallel algorithm for thinning digital patterns"* presented a parallel algorithm for skeletonizing different types of digital patterns. The algorithm is divided into two sub iterations that remove the boundary and corner points of the digital patterns. The first sub iteration aims at deleting the south-east boundary points and the north – west corner points, the second sub iteration is aimed at deleting the north- west boundary points and the south-east corner points. After several iterations only a skeleton of the pattern remains.

Ben K. Jang et al.[5] in their paper *"One pass parallel thinning : analysis, properties and quantitative evaluation"* defined Thinning as a procedure to transform a digital pattern, say, a connected component , to a connected skeleton of unit width. The one-pass parallel algorithm proposed requires only a single pass per iteration and uses a set of 5×5 templates. However, because of the limitation of the square-grid representation, resulting skeleton do not always closely approximate their corresponding medial axes. A number of other important issues, such as medial axis presentation, noise sensitivity, and over shrinking have been addressed. A set of measures to evaluate the proposed Thinning algorithm are also discussed. The Thinning algorithm is further extended to the derived grid for an isotropic medial axis representation.

M.V.Nagendraprasad et al. [6] in their paper *"An improved algorithm for thinnng binary digital patterns"* presented a parallel algorithm which yield good results with respect to speed and connectivity. However during implementation, the algorithm involves a number of time consuming steps.

Louisa Lam et al.[7] in their paper, *"An evaluation of parallel thinning algorithms for character recognition"* mentioned that the Thinning algorithms have played important role in the preprocessing phase of OCR systems. They have considered the performance of different Thinning algorithms from a number of aspects that are pertinent to OCR, from commonly measured quantities such as CPU time to the less easily quantifiable effects on the recognition process.

Samira S. Mersal et al.[8] in their paper *"A new parallel thinning algorithm for gray scale image"*, presented a parallel algorithm, for Thinning of gray scale images 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.

Y.Y.Zhang et. al. [9] in their paper, *"A thinning algorithm with two- subiterations that generates one – pixel wide skeletons"* have mentioned that many vectorization algorithms require one pixel-wide lines as input. One cannot use some published algorithms for vectorizing if there are redundant pixels in skeletons. They defined the Thinning of a binary image as an application of an iterative procedure for removing locally removable points. A Thinning algorithm is perfect if it can generate one-pixel wide skeletons. They further mentioned that in parallel Thinning, pixels are examined for deletion based on the results of only the previous iteration. Therefore, these algorithms are suitable for implementation on parallel processors where the pixels satisfying a set of conditions can be removed simultaneously. Their algorithm used 3×3 neighborhood in which each iteration is divided into two sub iterations in which only a subset of contour pixels are considered for removal. At the end of each sub iteration, the remaining image is updated for the next sub iteration.

L.Heutte et al. [10] in there paper *"A structural/ statistical feature based vector for hand written character recognition"* described the application of structural features to the statistical recognition of hand written characters.

N.H.Han et al.[11] in their paper, *"An efficient fully parallel thinning algorithm"* addressed an efficient parallel Thinning algorithm based on weight values. The weight values of a black pixel are calculated by observing neighboring pixels, and it gives us an efficient way to decide whether the pixel is deleted or not. Owing to the weight values, the proposed algorithm uses only 3*3 templates. The algorithm examines only the elimination conditions corresponding to the weight values of boundary pixels and all elimination conditions will not be searched as most other parallel iterative Thinning algorithms. They further mentioned that the thinned image helps in the extraction of the important features such as end points, junction points, and connections from image patterns.

V. PRESENT WORK & OBJECTIVE

Problem Formulation

The reduction of image can eliminate some counter distortions while maintaining significant topological and geometric properties. In more practical terms, thin-line representations of elongated patterns would be more suitable for extraction of critical features such as end-points, junction-points, and connection among the components. The vectorization algorithms often used in pattern recognition tasks also require one-pixel-wide lines as input.

Therefore, in real world, we have to thin the various images like BMP images to minimize the data to be handled, therefore there is need to have a software which can thin the images. There is need to study the various aspects of Thinning concepts. Keeping in mind that our goal is to come up with a set of strokes representing our original image, it would be nice to have a way to preserve the orientation of our shapes while removing information that doesn't interest us. One possibility is to consider the skeleton of an object.

We approximate the skeleton with a Thinning filter which gradually removes pixels from the borders of objects until something that looks a lot like the skeleton remains[15]. This Thinning algorithm has two desirable features; the first is that it will not remove any pixel that would cause one object to become two disconnected objects. It is also sensitive enough to properly thin segments which are two pixels thick. Any good Thinning algorithm must favor one direction over another in order to avoid removing every pixel in search of the one-true-single-pixel skeleton. The problem of connectivity has been solved in this algorithm which was present in the previous algorithms.

In real world there is a need for Thinning of images due to following reasons:

- To reduce the amount of data required to be processed.
- To reduce the time required to be processed.
- Extraction of critical features such as end-points, junction-points, and connection among the components.
- The vectorization algorithms often used in pattern recognition tasks also require one-pixel-wide lines as input.
- Shape analysis can be more easily made on line like patterns.

We aim to achieve the following objectives:

- To give an alternative parallel Thinning algorithm.
- To visualize and compare the performance of given alternative algorithm in terms of connectivity for regional language numerals.

VI. JUSTIFICATION FOR PROBLEM STATEMENT

The different parallel Thinning algorithms give different results in terms of maintaining the connectivity and generating the spurious branches. When we implement the discussed alternative parallel Thinning algorithm we observe that it provides better connectivity of pixels in the thinned image for almost all the test images. In the proposed algorithm we apply the single template in each pass and the output of each pass is passed onto the next pass, the connectivity and one- pixel width is guaranteed.

VII. METHODOLOGY

We propose the following plan for carrying out the present work:

1. **To create a database of regional language numerals from different users.**
   We create a database of hand written regional language numerals by considering the data created by different persons so that results can be observed for large image database. The database of regional language numerals is shown in appendix-B.
2. **To visualize the outputs of the two parallel Thinning algorithms.**
   We implement the two parallel Thinning algorithms for BMP images using C language. The outputs of these two algorithms are visualized to examine the connectivity of pixels and generation of spurious branches.
3. **To give an alternative parallel Thinning algorithm**
   We discuss an alternative parallel Thinning algorithm and then implement the same using C language. The algorithm is discussed separately in section 4.5.
4. **To compare the performance features of alternative parallel Thinning algorithm.**
   We visualize the results of the alternative Thinning algorithm and compare with the results of the previous two algorithms.

REFERENCES

[1] R. Gonzalez and R. E. Woods, *"Digital Image Processing",* Prentice Hall, 2002.

[2] T. Y., Zhang and C.Y. Suen, *"A fast parallel algorithm for thinning digital pattern",* Comm. Of ACM, vol. 27, no. 3, pp. 236-239.

[3] W. H. Abdullah, A.O.N. Saleh and A.H.Morad, *"A preprocessing algorithm for hand-written character recognition",* pattern recognition letters 7,1988

[4] Ben K. Jang, Roland T Chin, *"One pass parallel thinning: analysis, properties and quantitative evaluation*", IEEE transactions on pattern analysis and machine intelligence, vol. 14, no.11, November 1992,pp. 1129-1140.

[5] M.V.Nagendraprasad , P. S. P. Wang, A. Gupta, *"An improved algorithm for thinning binary digital patterns",* IEEE transactions on pattern analysis and machine intelligence, vol. 14, no.11, September 1995, pp. 386-389.

[6] Louisa Lam, C. Y. Suen, *"An evaluation of parallel thinning algorithms for character recognition",* IEEE transactions on pattern analysis and machine intelligence, vol. 17, no. 9, November 1992,pp. 914-919.

[7] Samira S. Mersal, A. M. Darwish, *"A new parallel thinning algorithm for gray scale image"*.

[8] Y.Y.Zhang, P. S. P. Wang, *"A thinning algorithm with two- subiterations that generates one – pixel wide skeletons",* Proceedings of ICPR, 1996, pp.457-461.

[9] L.Heutte, T. Paquet, J. V. Moreau, Y. Lecourtier, C. Olivier, *"A structural/ statistical feature based vector for hand written character recognition"* pattern recognition letters 19,1998, pp. 629-641