A Comparative Research of Thinning Algorithms

Rupal K. Snehkunj
Assistant Professor
Shri Ramkrishna Institute of Computer Education &Applied Sciences
Sarvajanik Education Society, Athwagate
Surat, India

ABSTRACT
Thinning is imperative step of preprocessing stage for handwritten and printed character, fingerprint recognition, circuit diagrams, etc. Due to the propagation of thinning algorithms, the choice of algorithm for an application has become very difficult, and a researcher in this area is often faced with the question of which algorithm to use. The goal of this paper is to perform comparative analysis amongst two or more algorithms and examine its effects.

General Terms
Thinning Algorithm

Keywords
Good Thinning Algorithm, skeletonization, pattern matching process

1. INTRODUCTION
Dramatic advances in communication and technology, pilot to usage of automated technique in various sectors of business/profession/etc. Science devoted to identify things as well as person by automatic means, by various techniques among which one of the technique is pattern recognition.

"Pattern recognition is the research area that studies the operation and design of systems that recognize patterns in data. It encloses subdisciplines like pre-processing, which may include thinning, noise reduction, image enhancements and error correction; structural extraction, where global and local structures may be found; post-processing, where the structures are converted into a more useful format.

Thinning is imperative operation that removes selected foreground pixels from binary images. The thinning process reduces the width of pattern to just a single pixel. **After an image, is thinned it is easier to find special points** (i.e minutiae in fingerprint recognition) **which are useful in pattern matching process.** Thinning when applied to a binary image, produces another binary image as output. The output of thinning process is a line drawing representation of a pattern called a 'thinned image'. Figure 1 shows thinning of the character "2".

Due to the propagation of thinning algorithms, the choice of algorithm for an application has become very difficult, and a researcher in this area is often faced with the question of which algorithm to use. For this reason, it is proposed to analyze two or more thinning algorithms and to examine its effects.

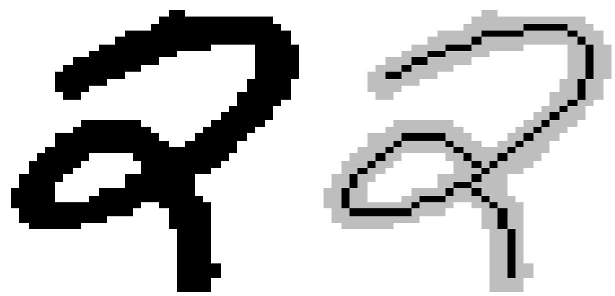

Figure 1: Thinning a handwritten "2".

2. THINNING ALGORITHMS

2.1 Importance of Thinning Process
In real world, there is an importance 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.

2.2 Structure of Thinning Algorithm
The basic structure of a thinning algorithm is given in Figure 2. This represents the one iteration step only. A thinning algorithm may consist of several iterations depending upon the logic. The deletion or retention of a (black) pixel 'p' would depend upon the configuration of pixels in the local neighborhood containing 'p'.

X_4	X_3	X_2
X_5	P	X_9
X_6	X_7	X_8

Fig. 2. Neighbors of a pixels p

Repeat

Remove deletable points from

the actual binary image

until no more points can be deleted.

Fig. 3. A thinning algorithm

According to the way we examine pixels, the thinning algorithms can be classified as 'Sequential' and 'Parallel'. In sequential algorithm, the pixels are examined for deletion in a fixed sequence in each iteration, and the deletion of a pixel p in the nth iteration depends on all the operations performed so far, i.e. on the results of (n-1)th iteration; as well as on the pixels already processed in (n)th iteration.

In a parallel algorithm, the deletion of pixels in the nth iteration depends only on the result of nth iteration; therefore, all the pixels can be examined independently in the parallel manner in each iteration.

Many thinning algorithms (or modifications of existing ones) have been proposed in recent years. Here two algorithms are discussed.

2.3 Criteria for Good Thinning Algorithms

Now, we have been aware that thinning is basically reducing a "thick" digital object to a "thin" skeleton. But what makes some thinning algorithms better than others? While the definition of "better" varies from application to application, we will discuss the quality of thinning algorithms that are generally considered superior. These include:

- sensitivity to noise
- preservation of topological and geometric properties
- isotropy
- reconstructability
- efficiency

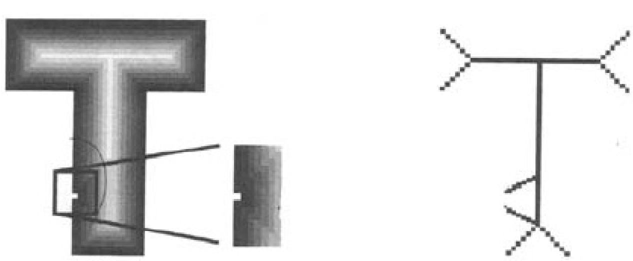

Figure 4 – Noise causing an incorrect output skeleton.

First, a thinning algorithm should be relatively insensitive to noise. Image noise is usually any pixels that are not part of the object currently under consideration. Noise can cause thinning algorithms to produce incorrect skeletons as shown in Figure 4.

Thinning algorithms should also preserve topological and geometric properties of the original object as much as possible. This includes connectedness of components, no spurious endpoints, and no excessive erosion of the original object.

Thinning algorithms should be isotropic, or rotation invariant. That is to say, the skeleton produced by the thinning algorithm for an object should be the same no matter which way the original object is rotated. Of course, depending upon the application, isotropy may not be a requirement.

Some applications require that the original pattern be reconstructable from the skeletonized version. This will affect algorithm overhead in a variety of ways depending upon the reconstruction method used.

And finally, a good thinning algorithm should be as efficient as possible. Application specifics have a large effect upon the importance of this factor. For example, real time applications such robot vision require fast, efficient thinning algorithms. In non-real time applications speed is less of a factor.

So is there a single thinning algorithm that does all of the above? The answer is no. As we will see the selection of a thinning algorithm usually involves trade-offs between one, or more of the above criteria.

3. THINNING ALGORITHMS ANALYSIS

3.1 A Fast Parallel Algorithm for Thinning Digital Patterns

It is fast and simple to be implemented. This algorithm is made by two sub-iterations. This algorithm is a parallel method that means the new value obtained only depend on the previous iteration value. In the first one, a pixel I (i , j) is deleted if the following conditions are satisfied:

1. Its connectivity number is one.
2. It has at least two black neighbours and not more than six.
3. At least one of I(i,j+1), I(i-1,j), and I(i,j-1) are white.
4. At least one of I(i-1,j), I(i+1,j), and I(i,j-1) are white.

In the second sub-iteration, the conditions in steps 3 and 4 change.

1. Its connectivity number is one.
2. It has at least two black neighbours and not more than six.
3. At least one of I(i-1,j), I(i,j+1), and I(i+1,j) are white.
4. At least one of I(i,j+1), I(i+1,j), and I(i,j-1) are white.

A 3x3 window is move down throughout the image and calculations are carried out on each pixel to decide whether it needs to stay in the image or not. To the right is a description of the window and the classification given to the pixels that surround the center pixel. The algorithm runs two subiterations continuously until the image reaches a stable state.

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 $(i+1, j+1)$

Pseudo Code for Zhang –Suen Thinning:

Let A(P) be the number of 01 patters in the order set P2 … P9.
Let B(P) be the number of non-zero neighbors of P
Do until image is stable (i.e. no changes made)
Sub-iteration 1:
Delete P from image if:
a) $2 \le B(P) \le 6$
b) A(P) = 1
c) P2 * P4 * P6 = 1
d) P4 * P6 * P8 = 1
Sub-iteration 2:
Delete P from image if:
a) and b) from above
c') P2 * P4 * P8 = 1
d') P2 * P6 * P8 = 1

At the end, pixels satisfying these conditions will be deleted. If at the end of either sub-iteration there are no pixels to be deleted, then the algorithm stops.

This processes effectively thins the image, however, it sometimes creates undesirably artifacts. In the example of a Zhang-Suen thinned fingerprint there are gaps between edges as well as regions of small area that need to be removed for proper regional analysis.

3.2 Preprocessing Algorithm for Hand-Written Character Recognition

This algorithm reduces the hand -written character into the unitary thin form. Each element is assigned the value '1' if it is covered by part of the character, and the value '0' otherwise. This algorithm involves two sub iterations [9]. In the first sub iteration the skeleton is scanned horizontally by the 3*4 pixels window shown. Any two points which are horizontally adjacent to each other and horizontally isolated from other points, are detected. With p1 and p4 representing terse two points, apply the following test whether one of them is redundant In the second sub iteration the thin is scanned vertically by the 4*3 pixel window shown. Any two points which are vertically adjacent to each other and vertically isolated from other points are detected.

P1 is deleted if one of the following conditions is true:

1. SP1 and p6=1:
2. SP2 and p2=1:
3. [(P2 and P3) or (P3 and P2 and P9)] and [(P5 and P6) or (P5 and P6 and P7)] Where SP1=P3 or P2 or P9. SP2= P6 or P5 or P7.

'and' and 'or' are logical 'AND' and logical 'OR' respectively.

If p1 is not redundant then p4 must be deleted if the following condition is not true: (P3 and P10) or (P5 and P12)

P_9	P_2	P_3	P10
P_8	P_1	P_4	P11
P_7	P_6	P_5	P12

Fig. 5 (a). A 3×4 pixel window.

P_9	P_2	P_3
P_8	P_1	P_4
P_7	P_6	P_5
P12	P11	P10

Fig. 5 (b). A 4×3 pixel window.

P_1 is deleted if one of the following conditions is true:

1. SP_{11} and p_4=1:
2. SP_{22} and p_8=1:
3. [(P_8 and P_7) or (P_7 and P_8 and P_9)] and [(P_4 and P_5) or (P_5 and P_4 and P_3)] Where SP_{11} =P_9 or P_8 or P_7, SP_{22}= P_3 or P_4 or P_5,

'and' and 'or' are logical 'AND' and logical 'OR' respectively.

If p_1 is not redundant then p6 must be deleted if the following condition is not true: (P_7 and P_{12}) or (P_5 and P_{10}).

4. PERFORMANCE EVALUATION

The algorithms are chosen for their significance and representation of different modes of operation in parallel thinning. The performance of these thinning algorithms can be evaluated on the basis of following parameters:

- Convergences of the thinned image to a unit width skeleton.
- Connectivity.
- Spurious branches
- Processing time.

The different parallel thinning algorithms give different results in terms of maintaining the connectivity and convergence to one pixel width. The two parallel thinning algorithms discussed when applied provide the results as shown in Figure 6. The two parallel thinning algorithms when implemented results in following pixel connectivity.

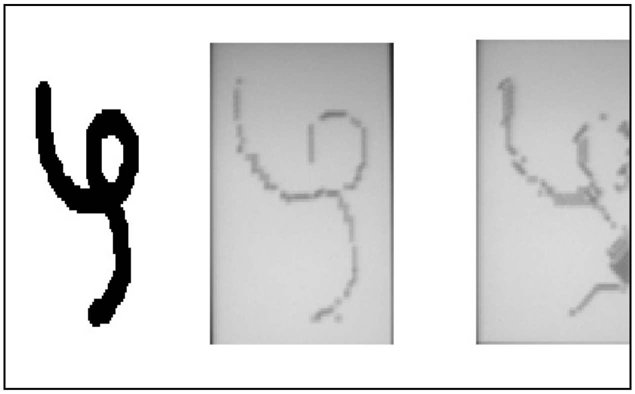

Fig.6. (a) Original numeral (b) Output of Algorithm 1 (c) Output of Algorithm 2

5. CONCLUSION AND FUTURE WORK

Output of Algorithm 1 is satisfying as compare to output of algorithm 2. But not fulfill expected results.

Switching from one technique to other is not the permanent solution. So instead of it, making changes in the current technique it is more fruitful by modifying in current algorithms or adding other new features. I would like to enhance the research work in these above techniques by adding new features to it.

6. REFERENCES

[1] Zhang T. Y, Suen C. Y, "A Fast Parallel Algorithm for Thinning Digital Patterns". Communications of the ACM. , Pages 236 - 239

[2] Selim Aksoy, Bilkent University, Department of Computer Engineering, RETINA Pattern Recognition Tutorial, Summer 2005

[3] Anil K. Jain, etc., "Statistical Pattern Recognition: A Review," IEEE Tran. on Pattern Analysis and Machine Intelligence, vol 22, No 1.

[4] Probability Refresher, notes for EE E6880, Statistical Pattern Recognition, Vittorio Castelli, Spring

[5] J. ANTHONY MOVSHON*, EDWARD H. ADELSON**, MARTIN S. GIZZI*and WILLIAM T. NEWSOME “The Analysis Of Moving Visual Patterns”

[6] O’Gorman L., “Overview of fingerprint verification technologies,” Elsevier Information Security Technical Report, Vol. 3, No. 1