Title: Thinning of binary images by the relaxation technique

Supervisor: Dr. Michael M.Y. Chang

By Woo Hok Luen

Student ID#: 90105130

Contents

Abstract

A parallel thinning algorithm is proposed to find the skeleton of a binary image by the relaxation technique. The aim of the process is to label each nonbackground pixel of the image as skeletal or nonskeletal. 4-neighbour distance transformation is used to assign initial probabilities for the pixels and give rise to the increment of the skeletal probability of each pixel. Pixels lying near the boundary tend to enhance the skeletal probabilities of the pixels lying farther away from the boundary. To cater for connectivity preservation for the image, the increment of the nonskeletal probabilities are found according to the point type of the pixel together with the 8-connectivity number. This decision process is consistent with most of the templates used in the SPTA proposed by Naccache and Shinghal. The thinning process is terminated and thus the skeleton is obtained when all nonbackground pixels have either high skeletal probability or high nonskeletal probability values (approaching 1). The experimental results show that the algorithm gives good skeletons for thinning thick images.

Ch. 1 Introduction

Thinning is an important stage in image processing. It has the following advantages:

1) It can reduce the volume of data for image representation, both for storage and retrieval.
2) It can facilitate the recognition process such as
   - finger print recognition
   - OCR
   - chromosome analysis

   etc.
3) It can simplify the computational procedure required for image description and classification.

Thinning usually refers to the process of continuously removing outer pixels until the final curves obtained are of one-pixel width. Since the result is called the skeleton of the original image, thinning sometimes is termed skeletonization. In some thinning algorithms, the result is obtained by finding the medial axis or symmetric axis of the image and as a result some people call this process the medial axis transformation.

Ch. 2 Review

2.1 Definition of a skeleton

R. W. Smith [1] pointed out that some researchers do not attempt to define the result of the thinning process or give a definition to the resultant skeleton but just iteratively strip or peel off the outer pixels while preserving the connectivity of the image.

On the other hand, Montanari [2] defined the skeleton by the process of wavefront propagations. In this definition, wavefronts are propagating from the edge of the original image towards the inside of it. When the wavefronts propagating from different edges of the image meet, the locus of their intersection is then the skeleton of the image. Based on the same idea, Yun Xia [3] developed an skeletonization algorithm by considering the fire front's propagation in digital binary shapes in which the skeleton is the result of the set of quench points obtained.

Rosenfeld [4] searched for the discs of maximal radii contained in the image and defined the set of the centers of the maximal discs as the skeleton. This kind of algorithm has an advantage in that the original image can be reconstructed from the skeleton so obtained. However this skeleton may not preserve the topological properties of the image, i.e. the skeleton for a connected image may not be connected.

Later, Davies and Plummer [5] defined a skeleton by incorporating additional points to the set of centres found by Rosenfeld's definition to achieve a connected result.

2.2 Features of a skeleton

Most of the thinning algorithms are iterative and pixels are examined and deleted if they meet certain conditions during each iteration. However, the question is: what are the essential features that a skeleton should possess? N. J. Naccache and R. Shinghal [6] gave the following as the essential properties for a skeleton:

1) The end points (including single points) of the original image are preserved.
2) The topological properties of the image should be preserved (i.e. the connectedness of the original image is kept unbroken).
3) The thinning algorithm should not bring about excessive erosion.

Different thinning algorithms proposed by different researchers are different only in the manners in which tests for meeting the above requirements for the resulting skeleton are conducted.

2.3 Parallel and sequential algorithms

When thinning algorithms are implemented according to the above mentioned criteria, there exist two different kinds of algorithms, namely parallel and sequential algorithms [7]. When the pixels are examined in parallel according to the result in the previous pass, the algorithm is a parallel one. A parallel algorithm has the following characteristics:

1) It needs more memory.
2) It is suitable for parallel processors.
3) It generates a mutual exclusion problem (the removal of the whole stroke in 2-pixel width).

Naccache and Shinghal reviewed 14 parallel thinning algorithms and proposed a Safe-point thinning algorithm (SPTA) [6] for thinning binary patterns. SPTA is a parallel thinning algorithm based on safe-point tests. Another example of parallel algorithms was that proposed by Shang and Suen [8].

Sequential algorithms are algorithms in which pixels are examined in sequence according to the most updated results [9]. Sequential algorithms use less memory than parallel ones and they do not have the mutual exclusion problem. Contour tracing and chain codes are commonly adopted in sequential algorithms.

2.4 Distance transformations

Many thinning algorithms involve the method of distance transformation (DT). Using a DT in the thinning algorithms has a number of advantages:

1) The computation of the DT is simple.
2) The computation of the DT is independent of the complexity of the boundary of the image.
3) It is possible to label the skeletal pixels directly from the result of a DT.
4) It is suitable to apply a DT on a cost-performance basis and can give more accurate and smoother results.

Some common DT's include city-block, chessboard, hexagonal, quasi-Euclidean and Euclidean [10]-[11]. Among these DT's, the city-block and chessboard are easier to compute and good for connectivity consideration but they have lower accuracy. On the other hand, hexagonal, quasi-Euclidean and Euclidean have higher accuracy and are able to provide smoother result but for their applications, connectivity has to be checked cautiously. Usually after the distance transformation some pixels (satisfying a certain number of requirements) need to be added to the set of local maximal pixels so that a connected skeleton can be produced [12]-[13].

2.5 Relaxation labelling process

Relaxation processes are a kind of algorithms for object labelling [14]. Suppose that a set of nodes is to be associated with a set of labels and the variable $P_i(\lambda)$ indicates whether the label λ is associated with the node i, then

$$P_i(\lambda) = \begin{cases} 1 & \text{if } \lambda \text{ is associated with node i} \\ 0 & \text{if } \lambda \text{ is not associated with node i} \end{cases}$$

where $0 \leq P_i(\lambda) \leq 1$ for all i, λ

$$\sum_{\lambda=1}^{m} P_i(\lambda) = 1 \qquad \text{for all } i = 1, \ldots, n.$$

The variables are then continuously updated by

$$P_i(\lambda) = \frac{P_i(\lambda)[1+Q_i(\lambda)]}{\sum_{k=1}^{m} P_i(k)[1+Q_i(k)]} \qquad \text{for all } i = 1, \ldots, n$$

and $Q_i(\lambda)$ is defined according to some updating rule. Due to its characteristic, $P_i(\lambda)$ can be treated as the probability that node i is associated with the label λ.

When thinning is treated as a labelling process, its objective is to classify each image pixel into skeletal or nonskeletal [15]-[16].

Ch. 3 The proposed algorithm

3.1 Definitions and notions

For a binary image, there are two kinds of pixels, 0-pixels and 1-pixels. Without loss of generality, we assume that all pixels on the first and last row or column are 0-pixels. The 0-pixels are called background pixels while the 1-pixels are called nonbackground pixels. Some refer them to white points and dark points respectively. For the sake of convenience, we list below some definitions and notions used in this paper.

1) Neighbours: The eight adjacent pixels to a pixel p are called the 8-neighbours of p, denoted by n_j, j = 0, 1, ..., 7 (Fig. 1). The neighbours n_0, n_2, n_4, n_6 are also called the 4-neighbours of p.

n3	n4	n5
n2	p	n6
n1	n0	n7

Fig. 1 The 8-neighbours of the pixel p.

2) Distance: For two pixels located at (x, y) and (r, s), the city-block distance between the pixels is defined as $|x - r| + |y - s|$ while the chessboard distance between them is defined as $\max(|x - r|, |y - s|)$.

3) Distance Transformations: The 4-neighbour DT of a binary image is to assign a value, called d4 value, to each pixel of the image according to the city-block distance. For the pixel (x, y), its d4 value can be found as

$$d_4(x,y) = \begin{cases} 0 & \text{if } (x, y) \text{ is a background pixel} \\ n & \text{if } (x, y) \text{ is a nonbackground pixel} \end{cases}$$

where n is the city-block distance between the pixel (x, y) and its nearest background pixel.

(a)

```
    11111
  111111111
 1111111111
11111   1111
111111111111
11111111111
11111111111
1111
1111        111
11111     1111
111111111111
  111111111
    11111
```

(b)

```
000000000000000
000001111100000
000112222211000
001222111221000
012321000122100
012332111232100
012332222221000
012321111111000
012210000000000
012210000001110
012321000012210
011222111122100
000112222211000
000001111100000
000000000000000
```

Fig. 2 The 4-neighbour DT for the binary image 'e'. (a) The original binary image. (b) The d4 value of each pixel of the image.

The 8-neighbour DT is defined in a similar way and the d_8 value for the pixel (x, y) can be assigned as

$$d_8(x,y) = \begin{cases} 0 & \text{if } (x, y) \text{ is a background pixel} \\ n & \text{if } (x, y) \text{ is a nonbackground pixel} \end{cases}$$

where n is the chessboard distance between the pixel and its nearest background pixel.

4) Connectedness: A pixel p and any one of its 8-neighbours are said to be 8-connected whereas p and any one of its 4-neighbours are 4-connected. Two nonbackground pixels are said to be 8-connected if there exists a chain of nonbackground pixels between them such that all successive pairs of pixels in this chain are 8-neighbours to each other. Similarly two nonbackground pixels are said to be 4-connected if there exists a chain of nonbackground pixels between them such that all successive pairs of pixels in the chain are 4-neighbours to each other. For most of the binary images, 8-connectivity should be preserved for the nonbackground pixels while 4-connectivity should be preserved for the background pixels.

5) Edge point and internal point: When a nonbackground pixel has at least one background pixel as its 4-neighbours, it is called an edge point. Otherwise, it is called an internal point.

6) End point: An end point is an edge point with only one nonbackground 8-neighbour.

7) Single point: In the original image, an edge point which has all 8-neighbours as background pixels is called a single point. If a pixel has the same property after the removal of all its 8-neighbours during the thinning process, it is not treated as a single point.

8) 8-connectivity number: For an edge point, the 8-connectivity number C8 is defined as [17]

$$C8 = \sum_{k=0}^{3} (n'_{2k} - n'_{2k} n'_{2k+1} n'_{2k+2})$$

where $n'_k = 1 - n_k$, $n_k \in \{0,1\}$ and $n_8 = n_0$.

The C8 of an edge point indicates the number of 8-connected components that are composed of the nonbackground 8-neighbours of it.

9) Simple point: When the 8-neighbours of a nonbackground pixel only form one simple 8-connected component, i.e. C8 = 1, then it is called a simple point.

10) Connection point: When the number of 8-connected components formed by the 8-neighbours of a pixel p is greater than one, p is called a connection point. Yu and Tsai [11] pointed out that the final result of any thinning algorithm should consist of only single points, end points and connection points.

11) Window: For any pixel p, the window of size n (or n by n window) centered at p is the set of p and its 8-neighbour pixels. If p is located at (x,y), then the n by n window centered at p is the set of pixels located at

$$\{ (i, j) \mid i = x-s, x-s+1, ..., x, ..., x+s-1, x+s; \quad j = y-s, y-s+1, ..., y, ..., y+s-1, y+s \}$$

where s is the integral part of n/2 and n is a positive odd integer.

Fig. 1 shows a 3 by 3 window centered at p.

3.2 The thinning problem

Thinning can be considered a labelling process in which each nonbackground pixel in the image is to be classified into two categories, namely skeletal or nonskeletal. Let

λ denote the skeletal class;

λ' denote the nonskeletal class;

$P_{x,y}(\lambda)$ denote the probability that the pixel (x,y) belongs to the skeletal class;

$P_{x,y}(\lambda')$ denote the probability that the pixel (x,y) belongs to the nonskeletal class.

Then a relaxation procedure is defined to update the $P_{x,y}(\lambda)$ and $P_{x,y}(\lambda')$ for each pixel until all the skeletal pixels are identified and all nonskeletal pixels are deleted.

3.3 Assigning initial probabilities

In this paper, we propose to use the 4-neighbour distance transformation for the nonbackground pixels of the image. Each pixel is associated with a d4 value after the transformation. The initial skeletal probability of each nonbackground pixel is defined to be proportional to the d4 value of (x, y) as

$$P^0_{x,y}(\lambda) \propto d_4(x,y)$$

or $$P^0_{x,y}(\lambda) = kd_4(x,y)$$

where 1/k is greater than the maximum d4 value of all the pixels and is chosen to be half of the width of the image in this paper.

After the initial skeletal probabilities are determined, the initial nonskeletal probabilities are fixed as

$$P^0_{x,y}(\lambda') = 1 - P^0_{x,y}(\lambda)$$

3.4 Iteration schemes

In the iteration process the probabilities $P_{x,y}(\lambda)$ and $P_{x,y}(\lambda')$ for the pixel (x,y) are updated according to the point type of the pixel, the d4 value and its neighbours within the window.

A skeletal value is defined initially for each pixel in the image as follows,

$$S^0(x,y) = \begin{cases} 0 & \text{for background pixels} \\ 1 & \text{for nonbackground pixels} \end{cases}$$

In the r^{th} iteration, the skeletal value of the nonbackground pixels is updated as

$$S^{r+1}(x,y) = \begin{cases} 0 & \text{if } P^{r+1}_{x,y}(\lambda) < \theta_1 \\ 1 & \text{if } P^{r+1}_{x,y}(\lambda) \geq \theta_1 \end{cases} \qquad \theta_1 : \text{threshold for deletion}$$

When $S^{r+1}(x,y)$ becomes zero for an nonbackground pixel, it means that the pixel has been classified nonskeletal and $P_{x,y}(\lambda)$ will not be altered since then. The thinning process is stopped when no remaining pixels are classified nonskeletal.

3.5 Net increment of the skeletal probabilities

The net increment of the skeletal probability for each nonbackground pixel (x,y) is defined by the n by n window centered at it as follows

$$Q^r_{x,y}(\lambda) = \{ \sum_{window}[d_4(x,y)-d_4(neighbours)]S^r(x,y)\}\alpha$$

$$= \{\sum_{i=-s}^{s}\sum_{j=-s}^{s}[d_4(x,y)-d_4(x+i,y+j)]S^r(x,y)\}\alpha$$

where s is the integral part of n/2 and α is a constant to be chosen to ensure that $|Q^r_{x,y}(\lambda)| < 1$.

If d4(x+i, y+j) < d4(x, y), the contribution from that neighbour is positive because (x,y) is closer to the medial axis than (x+i, y+j). On the contrary, the contribution is negative if (x, y) is farther away from the medial axis than (x+i, y+j) since d4(x+i, y+j) is then greater than d4(x, y).

3.6 Net increment of nonskeletal probabilities

The net increment of the nonskeletal probability for each nonbackground pixel (x,y) depends on its point type.

Point type			**$Q^r(\lambda')$**
internal point			0
edge point	single point		$-\beta_2$
	simple point	end point	$-\beta_2$
		nonend point	β_1
	connection point		$-1.5\beta_2$

β_1 and β_2 are two positive constants where $|\beta_1|$ and $|1.5\beta_2| < 1$.

If the pixel is a simple point but not an end point, it is expected to be deleted eventually and hence the nonskeletal probability is increased. If the pixel is a single point, a simple but end point or a connection point, then it is expected to be retained and thus the nonskeletal probability is decreased. To reflect the importance of the connection points, their decrease in the net increment of nonskeletal probabilities are larger than those of single points and end points by a factor of 1.5. On the other hand, there is no information if the pixel is an internal point because it may be a skeletal pixel or a nonskeletal pixel and hence its nonskeletal probability is unchanged in this iteration.

The net increment of the nonskeletal probability of a pixel of nonzero S(x,y) can be summarized as in Fig. 3.

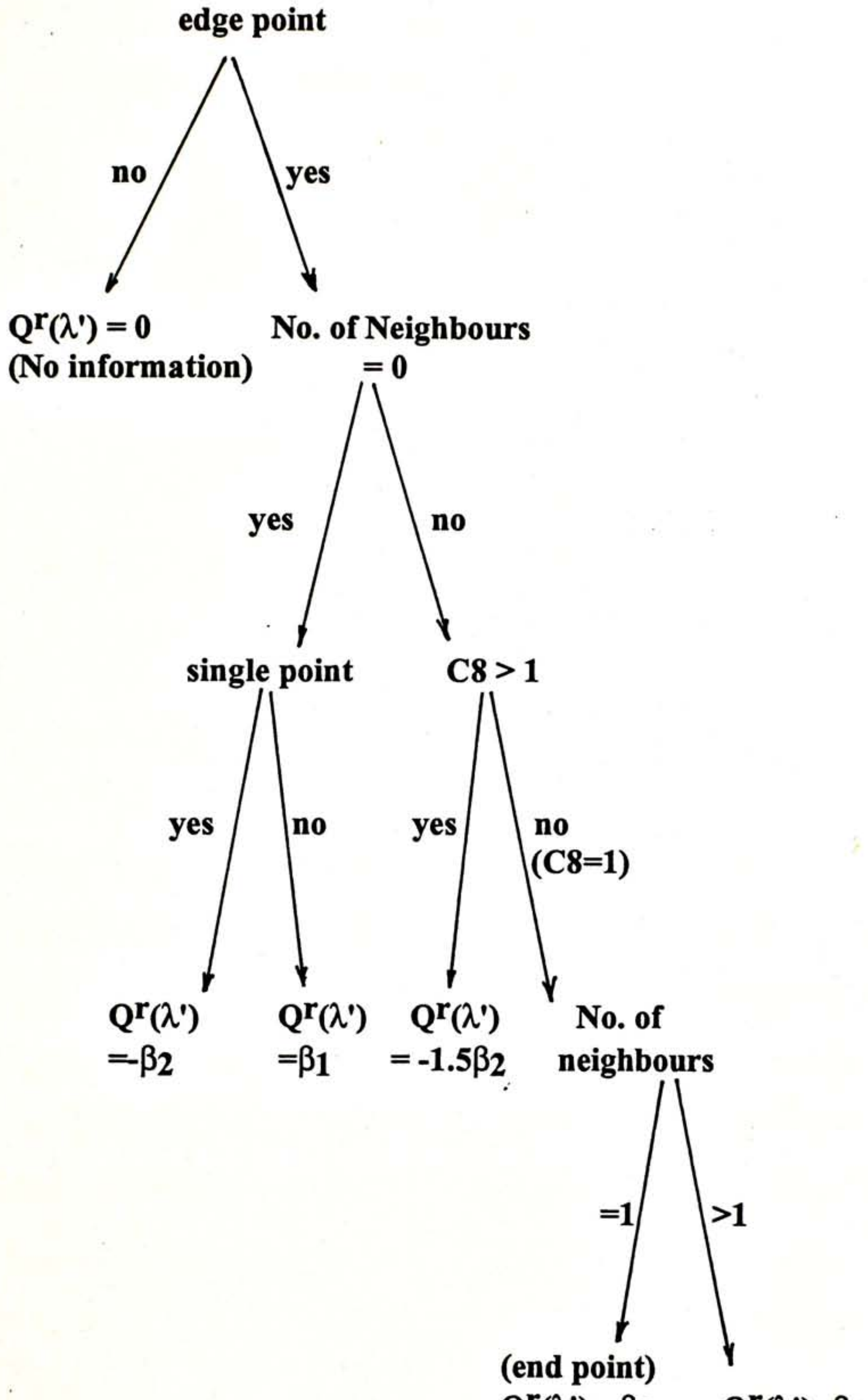

Fig. 3 The assignment of the net increment of the nonskeletal probability of a nonbackground pixel according to its point type.

3.7 Terminating condition

After the net increments of the skeletal and nonskeletal probabilities have been determined, the skeletal and nonskeletal probabilities are updated as

$$P^{r+1}(\lambda)=\frac{P^{r}(\lambda)[1+Q^{r}(\lambda)]}{P^{r}(\lambda)[1+Q^{r}(\lambda)]+P^{r}(\lambda')[1+Q^{r}(\lambda')]}$$

$$P^{r+1}(\lambda')=\frac{P^{r}(\lambda')[1+Q^{r}(\lambda')]}{P^{r}(\lambda)[1+Q^{r}(\lambda)]+P^{r}(\lambda')[1+Q^{r}(\lambda')]}=1-P^{r+1}(\lambda)$$

When the nonskeletal probability of a pixel (x,y) is greater than a certain threshold value ($1 - \theta_1$), the skeletal value $S^{r+1}(x,y)$ is set to 0, meaning that (x,y) is classified nonskeletal and deleted. On the contrary, if the skeletal probability is greater than the threshold ($1 - \theta_1$), then the pixel is considered a skeletal pixel. In either case, the probabilities of the pixel will not be altered in later iterations. For example, if θ_1 is chosen to be 0.05, then (x,y) is deleted when $P^{r}_{x,y}(\lambda') \geq 0.95$ (or $P^{r}_{x,y}(\lambda) < 0.05$). When $P^{r}_{x,y}(\lambda) > 0.95$ (or $P^{r}_{x,y}(\lambda') \leq 0.05$), (x,y) is regarded as a skeletal pixel.

A smaller θ_1 represents that a better thinning result is expected but a greater number of iterations will be needed (slower convergence). After all pixels have been classified either as skeletal or nonskeletal, the skeleton by the thinning algorithm is obtained.

Ch. 4 Experimental results

4.1 Parameters

In this algorithm, values have to be assigned to several parameters: k, n, α, β_1, β_2, θ_1.

The initial skeletal probability (and hence initial nonskeletal probability) of each nonbackground pixel depends on k the value of which has to be chosen carefully so that the initial probabilities cannot be greater than 1. Larger k values mean smaller initial skeletal probabilities for nonbackground pixels. k is chosen to be the reciprocal of half the width of the image pattern in this paper. For example, 1/k is equal to 20 for thinning a 40x40 image pattern.

The window size n and the constant α determine the number of neighbours and their contributions to the net increment of the skeletal probability of each pixel. There exists a restriction that $|Q^r_{x,y}(\lambda)| < 1$ and hence the values of n and α must meet this requirement. It has been found that a 3 by 3 window is appropriate for this algorithm.

β_1 and β_2 influence directly the choice and speed of pixel deletion. The function of β_1 is to increase the nonskeletal probability of a pixel while that of β_2 is to decrease the nonskeletal probability of that pixel. Hence when β_1 is smaller and β_2 is larger, the pixels will be deleted in a slower rate that implies that the result is better with the expense of the need of a larger number of iterations.

To enhance the convergence speed of the algorithm, a threshold value θ_1 is chosen so that when the skeletal and nonskeletal probabilities are greater than $(1 - \theta_1)$ they are essentially taken as 1 and the probability updating processes are stopped for those pixels. If the skeletal probability of a pixel is greater than $(1 - \theta_1)$, the pixel is considered skeletal. If the nonskeletal probability of a pixel is greater than $(1 - \theta_1)$, the pixel is regarded as nonskeletal and is deleted.

The values of the above parameters in this paper are summarized as follows:

Parameters	*Values*
β_1	0.35
β_2	0.45
n	3
α	0.10
θ_1	0.02

4.2 Results

With the values chosen as in the table, some binary patterns thinned by this algorithm are shown in the following figures. Fig. 4 shows the process of pixel deletion for a binary image 'e' while the skeletons of a number of binary images are shown in Fig. 5 - Fig. 9.

(a)

```
   11111
 111111111
1111111111
11111   1111
111111111111
11111111111
11111111111
1111
1111        111
11111      1111
1111111111111
  111111111
    11111
```

(b)

```
     11111 1
   111    11
  111      11
  1111    111
  111111111
  111      1
  11
  11         1
  111       111
1 111      11
     11111
```

(c)

```
   11111
 111    11
11        11
1111    111
111111111
11
11
11
111        11
 111      11
   11111
```

(d)

```
   11111
 111    11
11        11
 11     111
 11111111
11
11
11
111        11
 111     11
   11111
```

(e)

```
   11111
 111    11
 1        1
 11      11
 11111111
11
1
1
11        11
 111     11
   11111
```

(f)

```
     -----
   --*****--
  --*------*-
--*--    --*-
--**------*--
--*******--
--*--------
-*--
-*--           ---
--*--        -**-
---*------**-
  --*****--
    -----
```

Fig. 4 Thinning of the binary image 'e' by the proposed algorithm. (a) The original image. (b) The pixels remained after 5 iterations. (c) The pixels remained after 10 iterations. (d) The pixels remained after 15 iterations. (e) The pixels remained after 20 iterations. (f) The final result obtained after 142 iterations. Each '-' represents a deleted nonbackground pixel and each '*' denotes a skeletal pixel.

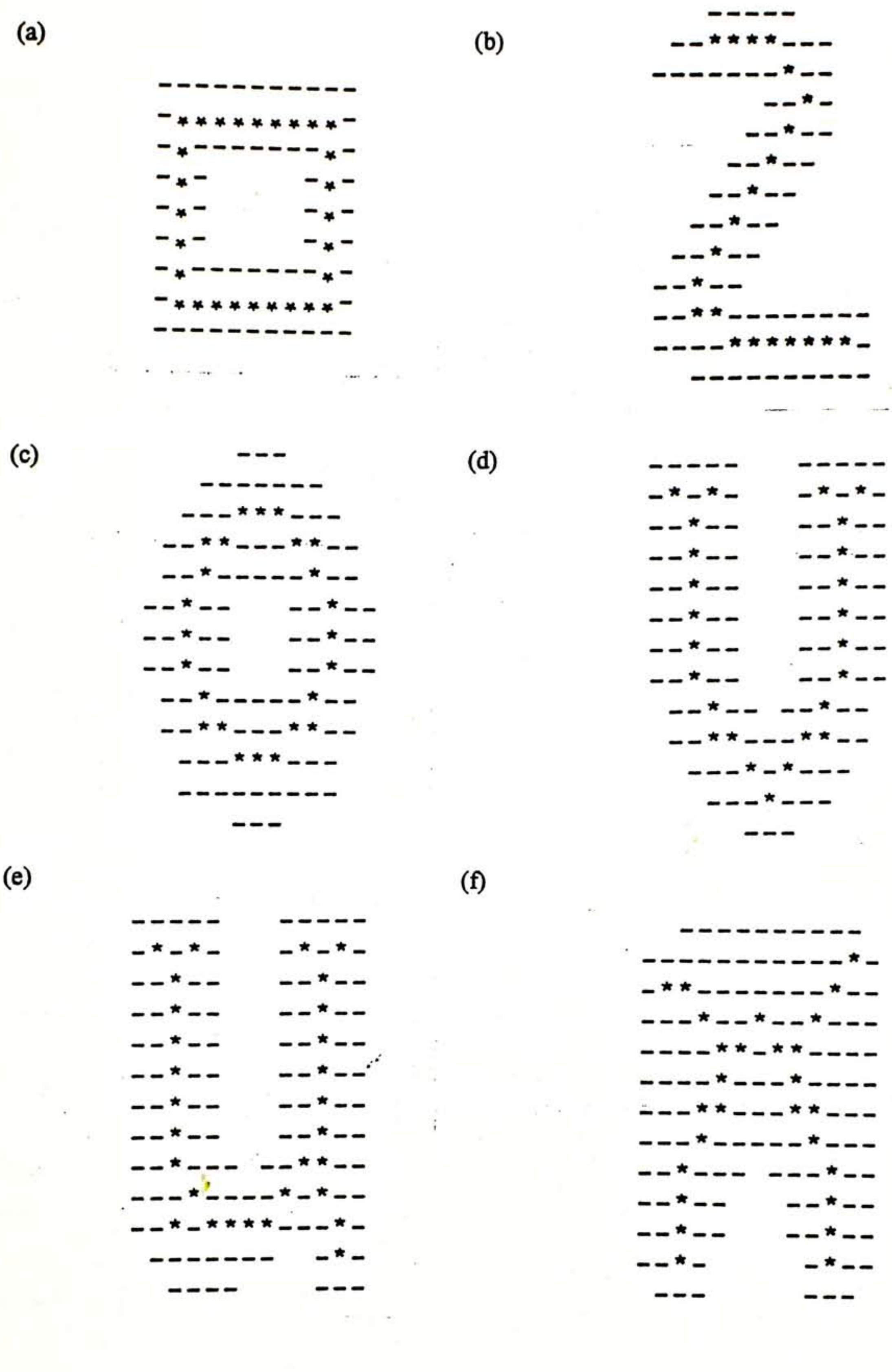

Fig. 5 The skeletons of some binary images obtained by the proposed thinning algorithm.

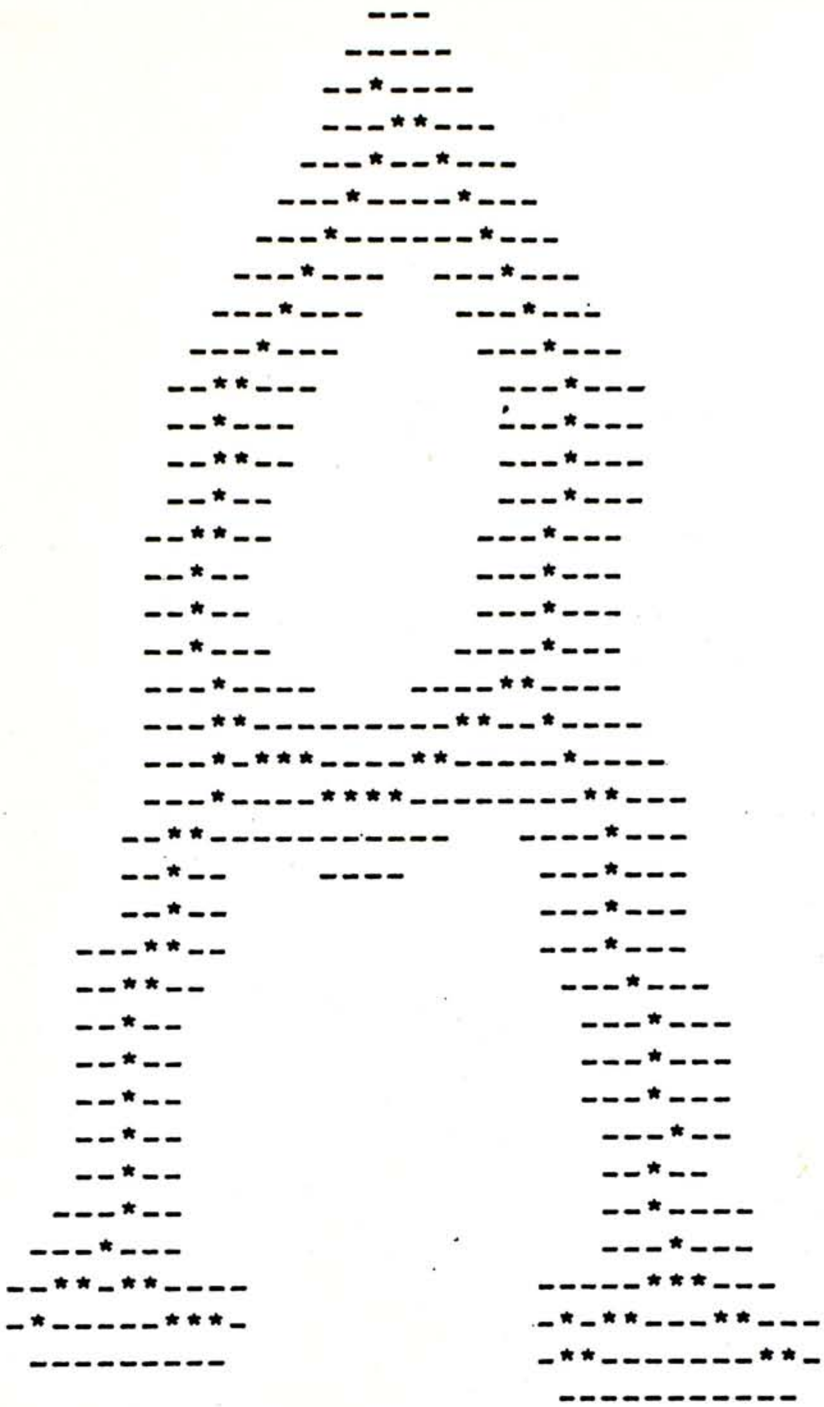

Fig. 6 The skeleton of the binary image 'A' obtained after 129 iterations.

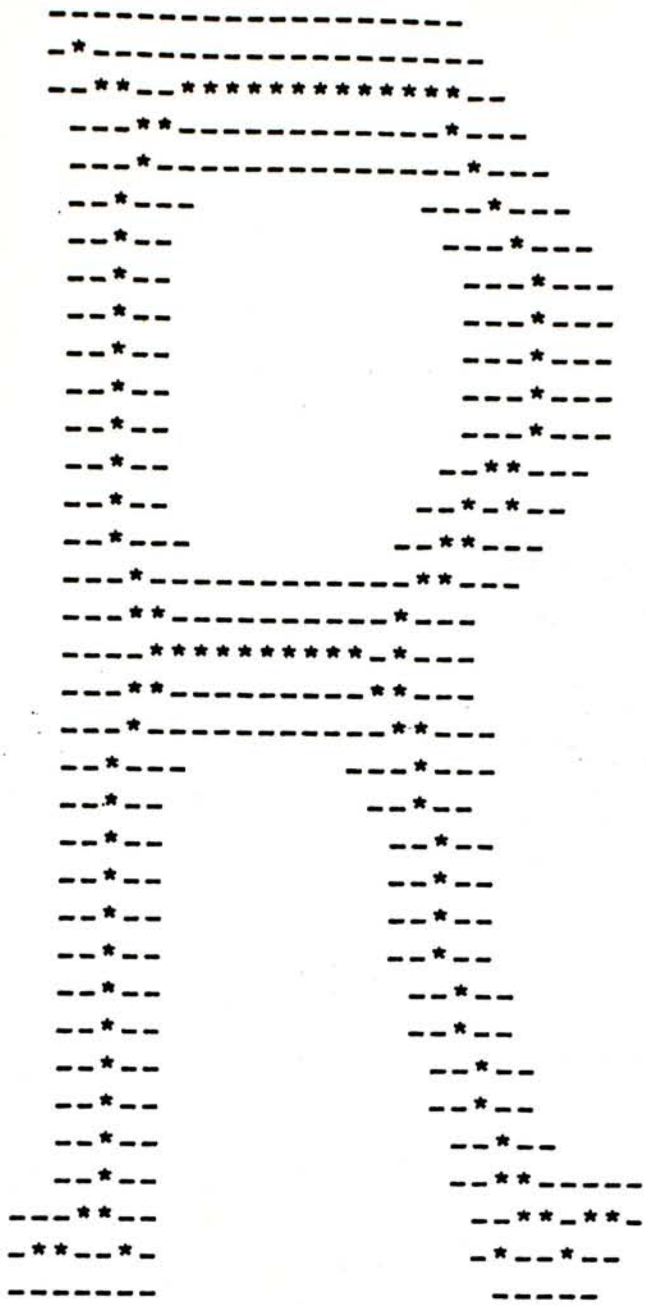

Fig. 7 The skeleton of the binary image '**R**' obtained after 136 iterations.

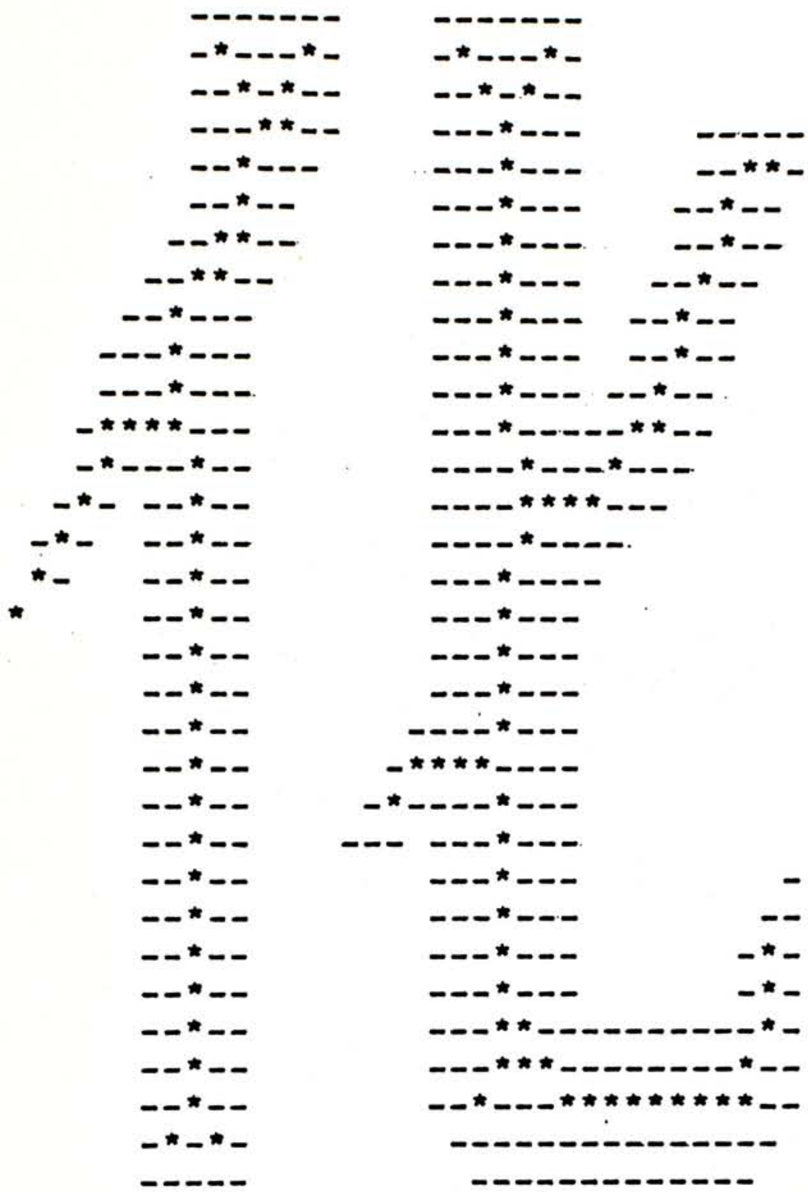

Fig. 8 The skeleton of the binary image '化' obtained after 129 iterations.

```
              ----
            -------
            -**--*--
            ---*****-
 ---       ---**----
 -*-       ---*---
 -*-        --*--
 -*-        --*--
--*--      ---*---
 --*--    ----*----
  --*-- ------*-------
   --*--------*--------
    --*-------*******--
     --******--------*---
      ------*---------*--
       -----*-----  --*--
        ----*----    --*--
         ---*---      --*--
         ---*---       -*-
         ---*---       -*-
         ---*---       -*-
         ---*---      -*-
        --**---      -*-
       -**--**--    ---
     --*------**--
   --**-    --*--
  -**--      --*--
 --*-        --*--
--*--         --*--
--*--          --*--
 --*--         --*--
  --*--         --*--
   --*--        --*--
    --*-         --*-
     -*-         -**-
      --         ----
```

Fig. 9 The skeleton of the binary image '🚶' obtained after 162 iterations.

Ch. 5 Discussion

5.1 4-neighbour DT and 8-neighbour DT

The 4-neighbour DT has been adopted in the proposed algorithm to implement the distance transformation for the image. When the 8-neighbour DT is used, the skeletons obtained are found to be inferior to those obtained by the 4-neighbour DT.

(a)

```
    -----
  --*****--
 --*-----*-
--*--   --*-
--**-----*--
--*******--
--*--------
-*--
-*--      ---
--*--    -**-
---*-----**-
  --*****--
    -----
```

(b)

```
     -
    -*-
   --*--
  ---*---
 ----*----
-*********-
 ----*----
  ---*---
   --*--
    -*-
     -
```

(c)

```
    -----
  --*****--
 --*-----*-
--**-   -*-*
-*-------*-*
-*-*******-
-*---------
-**-
-**-      ---
-**--    ----
---*-----*--
  --*****--
    -----
```

(d)

```
     *
    -*-
   --*--
  --***--
 --*-*-*--
***********
 --*-*-*--
  --***--
   --*--
    -*-
     *
```

Fig. 10 Comparisons are made between the 8-neighbour DT and 4-neighbour DT used in the proposed algorithm. (a) and (b) are images thinned by using the 4-neighbour DT while the same images thinned by using the 8-neighbour DT and the same parameters are shown in (c) and (d).

5.2 8-Connectivity number and safe-point tests

In the SPTA proposed by Naccache and Shinghal [6], safe-point tests are performed against a set of templates and the point will be flagged and deleted if the safe-point tests are satisfied. The templates for nonsafe points are listed as follows.

(a)	(b)	(c)	(d)
1 0 x	x x x	x 0 0	x x x
0 p x	0 p x	0 p 1	0 p 0
x x x	1 0 x	x 0 0	y y y

Fig. 11 The four templates used in the SPTA for non-safe point tests. x and y are used to denote a 'don't care' pixel.

It is found that the procedure used in determining the net increments of nonskeletal probabilities is consistent with those templates for which the point is to be retained (refer to the appendix) except for the following cases.

(a) W_1	(b) W_2	(c) W_3	(d) W_4
0 0 0	0 0 0	0 0 0	0 0 0
0 p 0	0 p 0	0 p 0	0 p 0
0 1 1	1 1 0	1 1 1	0 0 0

Fig. 12 The windows W_1 to W_4 that are considered as non-safe points in the SPTA have a net increase in the nonskeletal probability in the proposed algorithm (except when W_4 occurs in the original image).

In the SPTA, W_1 and W_2 are used to avoid excessive erosions while W_3 and W_4 are considered having no chance to occur in the original image but they retain shape information when happened in the intermediate stages. However, we allow W_3 and W_4 to exist in the original pattern and since we also have skeletal probability increment, so we let the net increment of nonskeletal probability be negative in the all these cases except for a W_4 occurring in the original pattern. In Fig. 13, comparisons are made for the image with and without the presence of a W_1 and W_3 in it.

(a)

```
  ------
 -**--*-
----**--
   --*--
   --*--
   --*--
   --*--
   --*--
   --*--
----***-----
-***---****-
------------
```

(b)

```
       *
  -----*
 -**--*-
----**--
   --*--
   --*--
   --*--
   --*--
   --*--
   --*--
----***-----
-***---****-
------------
```

(c)

```
-------
-*---*-
--*-*--
---*---
---*---
---*---
---*---
---*---
--*-*--
-*---*-
-------
```

(d)

```
   -
---*---
-*-*-*-
--***--
---*---
---*---
---*---
---*---
---*---
--*-*--
-*---*-
-------
```

Fig. 13 The difference between images in (a) and (b) is that a W_1 is present in (b). Similarly a W_3 occurs in (d). The results show that the algorithm can preserve the topological information of these windows.

5.3 Mutual exclusion problem

In some parallel algorithms, due to parallel examination of all pixels at the same time, the well-known mutual exclusion problem occurs [7]. As a result, a 2-pixel wide line may be completely removed because when pixels on both sides are examined simultaneously and independently, connectivity condition will not be violated and both of them can be deleted in the same pass.

One possible elimination of this problem is to introduce some sequential characteristics into the algorithm. Hence in certain thinning algorithms [6], [8] each pass is divided into a number of subiterations so that the symmetry can be broken.

Since our algorithm is a parallel one, we encounter the same difficulty which means that the algorithm may not give good result for an image having some parts that are too thin.

(a)

```
-------   *
-*****-   *
-*-----   *
-*-       *
-*-       *
-*-       *
-*-       *
-*-----   *
-*****-   *
-------   *
          *
```

(b)

```
-------   --
-*****-   --
-*-----   --
-*-       --
-*-       --
-*-       --
-*-       --
-*-----   --
-*****-   --
-------   --
          --
```

Fig. 14 The mutual exclusion problem occurs in the proposed algorithm and as a result the 2-pixel wide line in (b) is completely removed.

An alternative solution is to add a 1-pixel in between two adjacent 1-pixels to eliminate this problem. However, it brings about a trade off in the form of a larger number of iterations owing to the increase of pixels in the image to be thinned.

5.4 Skeleton not of unit width

When a binary image is thinned by using some algorithms, the skeleton obtained is not of unit width. The cross of a number of lines may not be a single point and interior points may exist in the skeleton [3], [17]. In the wavefront propagating type algorithms, the wavefronts may meet at a curve of 2-pixel width. This is a common problem to many algorithms and it happens also to the proposed algorithm.

(a)

```
--------
-*----*-
--*--*--
---**---
---**---
---**---
---**---
---**---
--*--*--
-*----*-
--------
```

(b)

```
---   ---   ---
-*-   -*-   -*-
--*---*---*--
  -*--*--*-
  --*-*-*--
-----***-----
-***********-
-----***-----
  --*-*-*--
  -*--*--*-
--*---*---*--
-*-   -*-   -*-
---   ---   ---
```

Fig. 15 Some skeletons are not of unit width. The medial axis is 2-pixel wide in (a) and there exists a cross of a number of lines in (b).

Separate algorithms for reducing the width of the skeleton from 2 pixels to 1 pixel were proposed by some researchers [18].

5.5 Development

This algorithm is a parallel one using the relaxation process and hence it is suitable to be implemented on parallel processing machines. Due to the progress in technology and reduction in hardware costs, this algorithm is promising because it is flexible and some kinds of correction terms, such as a contour term (containing boundary information of the original binary pattern) may be added without much difficulty to improve the thinning quality. Therefore the next development should be the introduction of a contour term to modify the skeleton so that it can reflect more correctly the boundary shape of the image to give a better thinned result.

Ch. 6 Conclusion

In this paper we propose a parallel thinning algorithm which applies the relaxation technique to locate the skeleton of a binary image. The function of updating the skeletal probabilities is to locate the medial axis of the image while the updating of nonskeletal probabilities tends to preserve the connectivity of the image.

The algorithm is suitable for application on a parallel processing machine. Because of its flexibility, it is possible for a modification term about the boundary information of the image to be added to give a better result. This algorithm has been shown capable of producing good skeletonization result for thick images.

Appendix

The 8-connectivity number and point type of the pixel p for the templates used in the SPTA are listed below. By symmetry, there are only three templates to be discussed.

Template 1:

x 0 0

0 p 1

x 0 0

1) <u>one 8-neighbour for p</u>

0 0 0

0 1 1

0 0 0

(C8=1 endpoint)

2) <u>two 8-neighbours for p</u>

1 0 0

0 1 1

0 0 0

(C8=2)

3) <u>three 8-neighbours for p</u>

1 0 0

0 1 1

1 0 0

(C8=3)

Template 2:

$$\begin{array}{ccc} 1 & 0 & x \\ 0 & p & x \\ x & x & x \end{array}$$

1) <u>one 8-neighbour for p</u>

$$\begin{array}{ccc} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{array}$$

(C8=1 endpoint)

2) <u>two 8-neighbours for p</u>

$$\begin{array}{ccc} 1 & 0 & 1 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{array} \qquad \begin{array}{ccc} 1 & 0 & 0 \\ 0 & 1 & 1 \\ 0 & 0 & 0 \end{array} \qquad \begin{array}{ccc} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{array}$$

(C8=2) (C8=2) (C8=2)

3) <u>three 8-neighbours for p</u>

$$\begin{array}{ccc} 1 & 0 & 1 \\ 0 & 1 & 1 \\ 0 & 0 & 0 \end{array} \qquad \begin{array}{ccc} 1 & 0 & 1 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{array} \qquad \begin{array}{ccc} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 1 & 0 \end{array}$$

(C8=2) (C8=3) (C8=3)

$$\begin{array}{ccc} 1 & 0 & 0 \\ 0 & 1 & 1 \\ 0 & 0 & 1 \end{array} \qquad \begin{array}{ccc} 1 & 0 & 0 \\ 0 & 1 & 1 \\ 0 & 1 & 0 \end{array}$$

(C8=2) (C8=2)

4) four 8-neighbours for p

1 0 1	1 0 1	1 0 1
0 1 1	0 1 1	0 1 1
0 0 1	0 1 0	1 0 0
(C8=2)	(C8=2)	(C8=3)
1 0 1	1 0 1	1 0 0
0 1 0	0 1 0	0 1 1
0 1 1	1 0 1	0 1 1
(C8=3)	(C8=4)	(C8=2)

5) five 8-neighbours for p

1 0 1	1 0 1	1 0 1
0 1 1	0 1 1	0 1 0
0 1 1	1 0 1	1 1 1
(C8=2)	(C8=3)	C8=3)

6) six 8-neighbours for p

1 0 1

0 1 1

1 1 1

(C8=2)

Template 3:

x	x	x
0	p	0
x	x	x

1) <u>one 8-neighbour for p</u>

0	0	0
0	1	0
0	0	0

(C8 =0 Singlepoint)

2) <u>two 8-neighbours for p</u>

0	0	0		0	0	0
0	1	0		0	1	0
1	0	0		0	1	0

(C8=1 endpoint) (C8=1 endpoint)

3) <u>three 8-neighbours for p</u>

0	0	0		0	0	0		0	0	1
0	1	0		0	1	0		0	1	0
1	1	0		1	0	1		1	0	0

(C8=1) (C8=2) (C8=2)

0	1	0		1	0	0		0	1	0
0	1	0		0	1	0		0	1	0
1	0	0		1	0	0		0	1	0

(C8=2) (C8=2) (C8=2)

4) <u>four 8-neighbours for p</u>

$\begin{matrix}0&0&0\\0&1&0\\1&1&1\end{matrix}$	$\begin{matrix}0&0&1\\0&1&0\\1&1&0\end{matrix}$	$\begin{matrix}0&1&0\\0&1&0\\1&1&0\end{matrix}$
($C8=1$)	($C8=2$)	($C8=2$)
$\begin{matrix}1&0&0\\0&1&0\\1&1&0\end{matrix}$	$\begin{matrix}0&0&1\\0&1&0\\1&0&1\end{matrix}$	$\begin{matrix}0&1&0\\0&1&0\\1&0&1\end{matrix}$
($C8=2$)	($C8=3$)	($C8=3$)

5) <u>five 8-neighbours for p</u>

$\begin{matrix}0&0&1\\0&1&0\\1&1&1\end{matrix}$	$\begin{matrix}0&1&0\\0&1&0\\1&1&1\end{matrix}$	$\begin{matrix}0&1&1\\0&1&0\\1&0&1\end{matrix}$
($C8=2$)	($C8=2$)	($C8=3$)
$\begin{matrix}1&0&1\\0&1&0\\1&0&1\end{matrix}$	$\begin{matrix}0&1&1\\0&1&0\\0&1&1\end{matrix}$	$\begin{matrix}1&1&0\\0&1&0\\0&1&1\end{matrix}$
($C8=4$)	($C8=2$)	($C8=2$)

6) <u>six 8-neighbours for p</u>

$\begin{matrix}0&1&1\\0&1&0\\1&1&1\end{matrix}$	$\begin{matrix}1&0&1\\0&1&0\\1&1&1\end{matrix}$
($C8=2$)	($C8=3$)

7) seven 8-neighbours for p

$$\begin{matrix} 1 & 1 & 1 \\ 0 & 1 & 0 \\ 1 & 1 & 1 \end{matrix}$$

($C8=2$)

Reference

[1] R. W. Smith, "*Computer processing of line images : A survey*". Pattern Recognition, Vol. 20, No. 1, p.7-15 (1987).

[2] U. Montanari, "*Continuous skeletons from digitized images*". J. Ass. Comput, March 16, p.534-539 (1969).

[3] Yun Xia, "*Skeletonization via the realization of the fire front's propagation and extinction in digital binary shapes*". IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 11, No. 10, p.1076-1086 (1989).

[4] J. L. Pfaltz and A. Rosenfeld, "*Computer representation of planar regions by their skeletons*". Communs. Ass. Comput. Mach., 10, p.119-125 (1967).

[5] E. R. Davies and A. P. N. Plummer, "*Thinning algorithms : a critique and a new methodology*". Pattern Recognition, 14, p.53-63 (1981).

[6] N. J. Naccache and R. Shinghal, "*SPTA: A proposed algorithm for thinning binary patterns*". IEEE Trans. on Systems, Man and cybernetics, Vol. SMC-14, No. 3, p.409-418 (1984).

[7] P. C. K. Kwok, "*A thinning algorithm by contour generation*". Communications of the ACM, Vol. 31, No. 11, p.1314-1324 (1988).

[8] T. Y. Zhang and C. Y. Suen, "*A fast parallel algorithm for thinning digital patterns*". Communications of the ACM, Vol. 27, No. 3, p.236-239 (1984).

[9] S. Suzuki and K. Abe, "*Sequential thinning of binary pictures using distance transformation*". IEEE Proc 8th ICPR, p.289-292 (1986).

[10] A. Rosenfeld and J. L. Pfaltz, "*Distance functions on digital pictures*". Pattern Recognition, Vol. 1, p.33-61 (1986).

[11] G. Borgefors, "*A new distance transformation approximating the Euclidean distance*". IEEE Proc 8th ICPR, p.336-338 (1986).

[12] C. Arcelli, L. P. Cordella and S. Levialdi, "*From local maxima to connected skeletons*". IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. PAMI-3, No. 2, p.134-143 (1981).

[13] A. Montanvert, "*Medial line : Graph representation and shape description*". IEEE Proc 8th ICPR, p.430-432 (1986).

[14] R. A. Hummel and S. W. Zucker, "*On the foundations of relaxation labelling processes*". IEEE Trans of Pattern Analysis and Machine Intelligence, Vol. PAMI-5, No. 3, p.267-287 (1983).

[15] S. S. Yu and W. H. Tsai, "*A new thinning algorithm for gray-scale images by the relaxation technique*". Pattern Recognition, Vol. 23, No. 10, p.1067-1076 (1990).

[16] C. H. Leung , Y. S. Cheung and Y. L. Wong, "*A knowledge-based stroke-matching method for Chinese character recognition*". IEEE Trans. on Systems, Man and Cybernetics, Vol. SMC-17, No. 6, p.993-1002 (1987).

[17] C. Arcelli and G. Sanniti di Baja, "*A width-independent fast thinning algorithm*". IEEE Trans. on Pattern Analysis and Machine Intelligence, Vol. PAMI-7, No. 4, p.463-474 (1985).

[18] H. Ogawa and K. Taniguchi, "*Thinning and stroke segmentation for handwritten Chinese character recognition*". Pattern Recognition, Vol. 15, No. 14, p.299-308, (1982).