A Comparative Study of Thinning Algorithms in Pattern Processing

Helen Ma

A Major Report

in

The Department of Computer Science

Presented in Partial Fulfillment of the Requirements
for the degree of Master of Computer Science at
Concordia University
Montréal, Québec, Canada

November 1983

ABSTRACT

A COMPARATIVE STUDY OF THINNING ALGORITHMS

IN PATTERN PROCESSING

Helen Ma

This report surveys the applications of thinning in image processing and pattern recognition, and examines the difficulties that confront existing thinning algorithms. A large number of existing thinning algorithms together with a new thinning algorithm designed by the author have been implemented on the Cyber 835 main frame computer at Concordia University. These algorithms are described and tested with a large variety of patterns. Each algorithm was verified with the original patterns published in the articles. Their characteristics are also presented and discussed. A systematic comparison of these algorithms has been done based on the following criteria : computation speed, memory requirement, connectivity after thinning, end-point retention, symmetry of medial line, and sensitivity to noise.

ACKNOWLEDGEMENTS

My thanks go to Mr. H. Tamura of the Electrotechnical Laboratory of Japan for his provision of programs and the assistance of Dr. S. Mori. Thanks also go to Dr. R. Shinghal and Mr. N. Naccache for their comments. Special thanks go to Dr. C.Y. Suen for his criticisms, suggestions and encouragement.

TABLE OF CONTENTS PAGE

INTRODUCTION

Thinning may be defined as the successive erosion of the outermost layers of a figure until only a connected unit framework or "skeleton" remains. This skeleton runs, ideally, along the medial lines of the limbs of the figure.

Thinning was originally devised to be used in optical character recognition [8,9]. However, since its concept was first introduced twenty years ago [1], thinning has assumed an important role in image processing and pattern recognition. It has been used as an aid in the inspection of printed circuit boards [2], the counting of asbestos fibres on air filters [3], the analysis of chromosome shapes [4], the examination of soil cracking patterns [5], the classification of fingerprints [6,7], and more recent applications such as digital facsimile transmission [10] and data reduction for map storage [11][12][13].

In general, thinning is employed in these applications for the following reasons: (a) to reduce the line images to medial lines of unit width in order to obtain topological information for further analysis, (b) to enable pictorial objects to be represented as simplified data structures, such as chain-coding [14], or (c) to reduce data storage for communication requirements.

Ultimately, all applications of thinning amount to the elimination of redundant information, leaving sufficient useful information to allow topological analysis and measurement of the shape, or in some cases regeneration of the original object[24]. However, different thinning algorithms were developed and employed in different applications. Lots of technical papers on this subject have been published in the past two decades. However, no clear standards have been developed such as methods of detecting one cell connectivity and stripping off the contour points; programming language used to implement these algorithms and the way to address the neighbouring points in a 3 x 3 window. Terminologies used in one algorithm may be referred differently in another, for example, edge point, contour point and border point have the same meaning, and there is no guarantee that the final thinning result is an ideal skeleton.

INHERENT PROBLEMS IN THE THINNING PROCESS

Three major problems must be overcome in order to obtain a good skeleton of a digitized image ---- (a) maintaining connectivity; (b) retaining end-points; and (c) ensuring that points are stripped off symmetrically. Both in principle and in practice these are real problems, and the solutions may yield conflicting results. Since thinning can be performed --- sequentially and in parallel, these problems are further complicated by the numerous possible ways of stripping off the points.

1) MAINTAINING CONNECTIVITY

In sequential algorithms, where each point in the picture is examined one by one in turn for a possible change, any modification (such as removal) will affect the subsequent processing of the remaining picture. Most sequential algorithms employ crossing number within a 3 by 3 window to achieve connectivity of the thinned pattern. A crossing number represents the condition of the neighbour points in the 3 X 3 window. However, in parallel algorithms, more complex formulae are employed to derive the crossing number to ensure connectivity. Nonetheless, some parallel algorithms choose to ignore such measures and they result in broken skeletons in certain situations [15]. Different algorithms attempt to solve the problem in different ways. Most of them appear to be "ad hoc", and in any case elaboration and detailed justification are not given by the authors.

2) RETAINING END-POINTS

As for connectivity of the thinned pattern, similar situations arise when we wish to assure that end-points are retained. For sequential algorithms, a main problem arises when the pattern contains horizontal and vertical branches which are two pixels wide, because the limbs tend to disappear altogether during the sequential stripping process. Some "ad hoc" procedure is usually invented to deal with this situation. The general approach is to locate the end point and try to avoid removing it from the pattern.

3) SYMMETRIC STRIPPING

A very complex procedure is required to ensure that points are stripped off symmetrically from a figure. Most sequential algorithms usually have a heavy bias to the lower right and this can be avoided by successive marking of border points in the stripping operation [23].

In this study, nine thinning algorithms are implemented in software. They are authored by :

1. E.S. Deutsch	[15]
2. C.J. Hilditch	[4]
3. H. Ma and J. Yudin	[16]
4. R. Stefanelli and A. Rosenfeld	[17]
5. D. Rutovitz	[18]
6. H. TAMURA	[19]
7. S. TSURUOKA	[20]
8. T.Y. ZHANG AND C.Y. SUEN	[22]
9. C. ARCELLI	[23]

These algorithms were chosen because the majority of them have been quoted widely in the literature. A large set of different types of binarized images are fed into the computer to test the performance of each algorithm. These 48 patterns are shown in appendix A. They include hand-written characters and numbers, Chinese ideograph, line-like particles, chromosome, and bubble chamber picture. The results are tabulated in terms of computation time, memory requirement, the conservation of structural information and noise immunity. In the following sections, an outline of each of these algorithms under test is presented together with some typical inputs and outputs. Discussions and comparisons of these algorithms are given in the concluding section.

THINNING ALGORITHM 1 ---

AUTHOR: E. S. DEUTSCH [15]

TERMINOLOGY:

4	3	2
5	A	1
6	7	8

a(i,j) element in row i and column j of a matrix.

r(k) value of pixel k in the 3 X 3 window

$k = 1, 2, 3, \ldots 8$

x crossing number is defined as

$$x = \sum_{k=1}^{8} | r(k+1) - r(k) |$$

where k has a period of 8 and x indicates the number of distinct 4 neighbour connected groups of black ('1') and white ('0') elements around a(i,j).

DESCRIPTION OF ALGORITHM:

Remove a black element ('1') if it satisfies all the following conditions:

1. $x = 0, 2$ or 4

2. $\sum_{k=1}^{8} r(k) \neq 1$

(that is, the element must have no neighbour or at least 2 in the pattern)

3. $r(1) \wedge r(3) \wedge r(5) = 0$

4. $r(1) \wedge r(3) \wedge r(7) = 0$

5. if $x = 4$, then in addition, either condition (a) or (b) must hold ---

(a) $\{ r(1) \wedge r(7) = 1 \}$ and
$\{ r(2) \vee r(6) = 1 \}$ and
$\{ r(3) \wedge r(4) \wedge r(5) \wedge r(8) \} = 0$

(b) $\{ r(1) \wedge r(3) = 1 \}$ and
$\{ r(4) \vee r(8) = 1 \}$ and
$\{ r(2) \wedge r(5) \wedge r(6) \wedge r(7) \} = 0$

The pattern is tested continuously until no further change occurs.

RESULT OF TESTING:

CPU TIME .... 27.896 seconds *

MEMORY REQUIREMENT .... 26,240 words **

CONSERVATION of TOPOLOGICAL INFORMATION .... 5 ***

NO. OF PATTERNS TESTED 48

* CDC CYBER/835 FORTRAN V, the CPU time refers to the total amount of time in seconds required by the computer to process all 48 test patterns.

** Memory requirement includes the two 2-dimensional arrays of 2 X (110 X 80) = 17,600 words to store the original pattern and the thinned pattern and other program buffers for the operations.

*** See explanation on p. 34.

THINNING ALGORITHM 2 ---

AUTHOR: C. JUDITH HILDITCH [4]

TERMINOLOGY:

window

```
+---+---+---+
| n4| n3| n2|
+---+---+---+
| n5| p | n1|
+---+---+---+
| n6| n7| n8|
+---+---+---+
```

a(i) value of pixel n in the 3 X 3 window

$i = 1,2,3,\ldots 8$

$f(n_i)$ member of the following sets:

I = Set of black elements on the pattern

R = Set of removed elements from the pattern ;

N = Set of non-removed elements

DESCRIPTION OF ALGORITHM :

A black point ('1') p is removed from the pattern if all the following conditions are satisfied.

1. Point lies on the edge of the pattern.

$u(p) >= 1$ if $u(p) = a(1) + a(3) + a(5) + a(7)$

where $a_i = 1$ if $f(n_i) \in N$ or $a_i = 0$

2. Point is not the tip of a thin line.

$v(p) >= 2$ if $v(p) = \sum_{i=1}^{8} (1-a_i)$

3. point is not the last remaining point of the pattern.

$w(p) >= 1$ if $w(p) = \sum_{i=1}^{8} c_i$ or $c_i = 0$

4. Removal of the point should not alter the connectivity of the result.

$X(p) = 1$ crossing number is defined as

$$X(p) = \sum_{i=1}^{4} b_i \text{ where } b_i = 1 \text{ if } f(n_{2i-1}) \in N$$

and either

$f(n_{2i}) \in I \cup R$ or $f(n_{2i+1}) \in I \cup R$

or $b_i = 0$, otherwise.

5. Removal of the point along with any one of its neighbours which has already been removed does not change the connectivity of the end result.

$f(n_i) \in R$ or $X_i(p) = 1$ $(i=1,2,3,...8)$

The above operation is repeated until p remains unchanged.

RESULT OF TESTING:

CPU TIME .... 20.401 seconds

MEMORY REQUIREMENT .... 26,496 words

CONSERVATION OF TOPOLOGICAL INFORMATION 2

NO. OF PATTERNS TESTED 48

THINNING ALGORITHM 3 ---

AUTHORS: H. MA AND J. YUDIN [16]

TERMINOLOGY:

Contour tracing -- find the leftmost black point(p). A scanning spot moves point by point, from bottom to top of the leftmost column and successively repeats the procedure on the column immediately to the right of the column previously scanned until a black point(p) is found.

Two preprocessing steps were done before the thinning process. One of them eliminates all isolated points which are black elements with no neighbour. The other step fills a gap or eliminates a notch on an edge.

DESCRIPTION OF ALGORITHM:

The thinning operation traces the contour of the image. It scans the image horizontally by passing a 3 X 3 window over each contour point and performs the tests described below.

[I] P will be retained if it satisfies any of the following conditions:-

1. BREAK POINT TEST

A point of interest is considered as a break point if there exists one or more black points in the positions covered with symbol "X" in a 3 by 3 window and match one of the following patterns :

```
(A) +---+---+---+   (B) +---+---+---+   (C) +---+---+---+
    | X | X | X |       | X | X | X |       | 1 | O | X |
    +---+---+---+       +---+---+---+       +---+---+---+
    | O | P | O |       | O | P | X |       | O | P | X |
    +---+---+---+       +---+---+---+       +---+---+---+
    | X | X | X |       | 1 | O | X |       | X | X | X |
    +---+---+---+       +---+---+---+       +---+---+---+

(D) +---+---+---+   (E) +---+---+---+   (F) +---+---+---+
    | X | O | X |       | X | O | 1 |       | X | X | X |
    +---+---+---+       +---+---+---+       +---+---+---+
    | X | P | X |       | X | P | O |       | X | P | O |
    +---+---+---+       +---+---+---+       +---+---+---+
    | X | O | X |       | X | X | X |       | X | O | 1 |
    +---+---+---+       +---+---+---+       +---+---+---+
```

2. LOOP POINT TEST

A point of interest is considered as a loop point if there exists a 3 by 3 window that matches one of the following patterns with symbol "X" representing a don't care point.

```
(a) +---+---+---+   (b) +---+---+---+
    | X | 1 | X |       | 1 | X | 1 |
    +---+---+---+       +---+---+---+
    | 1 | P | 1 |       | X | P | X |
    +---+---+---+       +---+---+---+
    | X | 1 | X |       | 1 | X | 1 |
    +---+---+---+       +---+---+---+
```

3. CORNER POINT TEST

A point of interest is considered as a corner point if there exists a 3 by 3 window that matches one of the following patterns with symbol "X" representing a don't care point and symbol "*" representing a contour point.

```
+---+---+---+   +---+---+---+   +---+---+---+   +---+---+---+
| X | X | X |   | X | * | X |   | X | * | X |   | X | * | X |
+---+---+---+   +---+---+---+   +---+---+---+   +---+---+---+
| * | P | * |   | * | P | * |   | * | P | X |   | X | P | * |
+---+---+---+   +---+---+---+   +---+---+---+   +---+---+---+
| X | * | X |   | X | X | X |   | X | * | X |   | X | * | X |
+---+---+---+   +---+---+---+   +---+---+---+   +---+---+---+
     (A)             (B)             (C)             (D)
```

[II] P will be removed if it satisfies at least one of the following conditions:-

1. ISOLATED POINT

A point of interest is considered as an isolated point if all 8 neighbours in a 3 by 3 window are zeros.

```
+---+---+---+
| 0 | 0 | 0 |
+---+---+---+
| 0 | P | 0 |
+---+---+---+
| 0 | 0 | 0 |
+---+---+---+
```

2. EXCESSIVE EROSION TEST

```
+---+---+---+         +---+---+---+
| 0 | 0 | 0 |         | 0 | 0 | 0 |
+---+---+---+         +---+---+---+
| 0 | P | p4|  ===>   | 0 | 0 | 1 |
+---+---+---+         +---+---+---+
| 0 | * | * |         | 0 | * | * |
+---+---+---+         +---+---+---+
```

P will be removed and point p4 will be changed to a black point.

3. ENDPOINT TEST

If the sum of the eight neighbours is 1, call this point B, and the 8 neighbours of point B will be examined. If the sum of all neighbour points of point B is equal to 2 then it will not be removed.

4. TESTS IN I.1 TO I.3 ARE NOT SATISFIED

This process of contour tracing and subsequent thinning are repeated until the pattern is nowhere more than one unit wide.

RESULT OF TESTING:

CPU TIME	23.438 seconds
MEMORY REQUIREMENT	26,752 words
CONSERVATION OF TOPOLOGICAL INFORMATION	O
NO. OF PATTERNS TESTED	48

THINNING ALGORITHM 4 ---

AUTHORS: R. STEFANELLI AND A. ROSENFELD [17]

TERMINOLOGY:

Contour point -- point for which at least one of the conditions shown in the following figures is satisfied.

```
+---+---+---+   +---+---+---+   +---+---+---+   +---+---+---+
| X | X | X |   | X | O | X |   | X | X | X |   | X | X | X |
+---+---+---+   +---+---+---+   +---+---+---+   +---+---+---+
| X | P | X |   | X | P | X |   | O | P | X |   | X | P | O |
+---+---+---+   +---+---+---+   +---+---+---+   +---+---+---+
| X | O | X |   | X | X | X |   | X | X | X |   | X | X | X |
+---+---+---+   +---+---+---+   +---+---+---+   +---+---+---+

  (A) LOWER       (B) UPPER       (C) LEFT        (D) RIGHT
```

where: 'x' denotes a don't care point which can be '1' or '0', but at least one of the don't care points must be '1', otherwise 'p' is an isolated point.

FINAL point -- point for which at least one of the conditions shown in the following figures is satisfied.

```
+---+---+---+   +---+---+---+   +---+---+---+   +---+---+---+
| X | X | X |   | O | X | X |   | Y | O | X |   | Y | Y | O |
+---+---+---+   +---+---+---+   +---+---+---+   +---+---+---+
| O | P | O |   | Y | P | X |   | Y | P | X |   | Y | P | X |
+---+---+---+   +---+---+---+   +---+---+---+   +---+---+---+
| Y | Y | Y |   | Y | Y | O |   | Y | O | X |   | O | X | X |
+---+---+---+   +---+---+---+   +---+---+---+   +---+---+---+

    (A1)            (A2)            (A3)            (A4)
```

where: at least one "X" and one "Y" must not be zero.

DESCRIPTION OF ALGORITHM:

A given contour point ('1') p will not be deleted but will be stored in a final point matrix if it satisfies the corresponding conditions.

(a) for LOWER contour point :

if it matches (A1) (as described in the final point definition) and satisfies (B1) and (C1).

(b) for UPPER contour point :

if it matches (A2) and satisfies (B2) and (C2).

(c) for LEFT contour point :

if it matches (A3) and satisfies (B3) and (C3).

(d) for RIGHT contour point :

if it matches (A4) and satisfies (B4) and (C4).

```
+---+---+---+   +---+---+---+   +---+---+---+   +---+---+---+
| X | X | X |   | X |   | O |   |   | 1 | O |   |   | O | X |
+---+---+---+   +---+---+---+   +---+---+---+   +---+---+---+
|   | P | O |   | X | P | 1 |   | O | P |   |   | 1 | P | X |
+---+---+---+   +---+---+---+   +---+---+---+   +---+---+---+
| O | 1 |   |   | X | O |   |   | X | X | X |   | O |   | X |
+---+---+---+   +---+---+---+   +---+---+---+   +---+---+---+
    (B1)            (B2)            (B3)            (B4)

+---+---+---+   +---+---+---+   +---+---+---+   +---+---+---+
|   |   |   |   |   |   |   |   |   | O | 1 |   | 1 | O |   |
+---+---+---+   +---+---+---+   +---+---+---+   +---+---+---+
| O | P |   |   |   | P | O |   |   | P | O |   | O | P |   |
+---+---+---+   +---+---+---+   +---+---+---+   +---+---+---+
| 1 | O |   |   |   | O | 1 |   |   |   |   |   |   |   |   |
+---+---+---+   +---+---+---+   +---+---+---+   +---+---+---+
    (C1)            (C2)            (C3)            (C4)
```

Each black element in the pattern will be tested and deleted if applicable until the modified pattern and the final point matrix are identical.

RESULT OF TESTING:

CPU TIME .... 49.653 seconds

MEMORY REQUIREMENT .... 29,696 words

CONSERVATION OF TOPOLOGICAL INFORMATION 13

NO. OF PATTERNS TESTED 48

THINNING ALGORITHM 5 ---

AUTHOR: D. RUTOVITZ [18]

TERMINOLOGY:

$R_{i,j}$ — coordinates of a point in a binary picture

$g(R_{i,j})$ — gray scale of the given picture point

$N_{i,j}(k)$ — the kth neighbour point of $R_{i,j}$

where $N_{i,j}(1) = R_{i+1,j}$

(right hand side neighbour) and it increases

in counter-clockwise direction with $N_{i,j}(9)=N_{i,j}(1)$

$Y(k)$ — gray scale of the kth neighbour (i.e. $g(N_{i,j}(k))$)

$X(R_{i,j})$ — cross number of the given picture point

defined as

$$X(R_{i,j}) = \sum_{k=1}^{8} |Y(k+1) - Y(k)|$$

$X(k)$ — cross number of the kth neighbour

DESCRIPTION OF ALGORITHM:

The transformation function value of $R_{i,j}$ is unchanged unless all of the following conditions are satisfied

1. $g(R_{i,j})=1$

2. $X(R_{i,j})=2$

3. $\sum_{k=1}^{8} Y(k) \geq 2$

4. Y(1)*Y(3)*Y(5) = 0 or X(3) not equal to 2

5. Y(1)*Y(3)*Y(7) = 0 or X(1) not equal to 2

in which case the transformation value becomes 0 (i.e. deleted from the original).

RESULT OF TESTING:

CPU TIME .... 30.005 seconds

MEMORY REQUIREMENT .... 28,096 words

CONSERVATION OF TOPOLOGICAL INFORMATION 4

NO. OF PATTERNS TESTED 48

THINNING ALGORITHM 6 ---

AUTHOR: H. TAMURA [19]

TERMINOLOGY:

4	3	2
5	X	1
6	7	8

S = { 1,2,3,...,8 }

S1= { 1,3,5,7 }

S2= { 2,4,6,8 }

Neighbourhood-

4-Neighbour of the element x $\{ x_k \mid k \leftarrow S1 \}$

8-Neighbour of the element x $\{ x_k \mid k \leftarrow S1 \}$

Connectivity-

Two elements a_1 and a_2 with common value are said to be 4-connected (8-connected) if there exists a sequence of elements $Y_1 (=a_1)$, Y_2, ..., $Y_n (=a_2)$ such that each Y_i is in the 4 neighbours (or 8 neighbours) of Y_{i-1} $(1<=i<=n)$ and all have the same value as a_1 and a_2. It should be noted that 8- connectivity must be adopted for 0-elements when 4- connectivity adopted for 1-elements and vice versa.

4-connectivity case

$$N_{c4} = \sum_{k<-s1} (Y_k - Y_k Y_{k+1} Y_{k+2})$$

8-connectivity case

$$N_{c8} = \sum_{k<-s1} (\bar{Y}_k - \bar{Y}_k \bar{Y}_{k+1} \bar{Y}_{k+2})$$

where $\bar{Y}_k$ means $1-Y_k$

Crossing number x-

$$x = \sum_{k<-s} |Y_k - Y_{k+1}|$$

indicates the number of distinct chains of blacks and whites in the neighbours of YO. To distinguish 4-connected branches and 8-connected branches, the author uses $x' = x/2$ to characterize the property of the test point

x' = 0 interior or isolated point

= 1 edge element

= 2 connecting element

= 3 branching element

= 4 crossing element

DESCRIPTION OF ALGORITHM:

A black point is removed under the following conditions:

1. $N_{c4} = N_{c8} = 1$

2. $N_{c4} = 0$ and $N_{c8} = 1$

3. $N_{c4} = 1$ and $N_{c8} = 0$

A black point(P) is retained if it satisfies at least one of the following configurations:

```
+---+---+---+   +---+---+---+   +---+---+---+   +---+---+---+
|   | 1 | 0 |   | 0 | 1 |   |   |   | Y |   |   |   | Y |   |
+---+---+---+   +---+---+---+   +---+---+---+   +---+---+---+
| Y | P | 1 |   | 1 | P | Y |   | 1 | P | Y |   | Y | P | 1 |
+---+---+---+   +---+---+---+   +---+---+---+   +---+---+---+
|   | Y |   |   |   | Y |   |   | 0 | 1 |   |   |   | 1 | 0 |
+---+---+---+   +---+---+---+   +---+---+---+   +---+---+---+

    (A1)            (A2)            (A3)            (A4)
```

Where blank points mean "don't care" conditions and at least one Y must be a black point.

The pattern is scanned horizontally and the black points ('1') are tested and deleted accordingly until no more points can be deleted.

RESULT OF TESTING:

CPU TIME .... 26.269 seconds

MEMORY REQUIREMENT .... 26,688 words

CONSERVATION OF TOPOLOGICAL INFORMATION 9

NO. OF PATTERNS TESTED 48

*NOTE:

This program was supplied by Mr. H. Tamura and modified by the author to run it on the Cyber 835 computer.

THINNING ALGORITHM 7 ---

AUTHOR : S. TSURUOKA [20]

TERMINOLOGY:

3 by 3 window

```
                    +---+
                    | B |
+---+---+---+---+---+
| A |p4 |p5 |p6 |
+---+---+---+---+
    |p3 | X |p7 |
    +---+---+---+
    |p2 |p1 |p8 |
    +---+---+---+
```

DESCRIPTION OF ALGORITHM:

When it is a black element ('1'), "X" is retained if any of the following conditions are satisfied :

1. p1 + p3 + p5 + p7 = 4
2. p4 = 1 ^ p3 = 0 ^ p5 = 0
3. p4 = 0 ^ p1 = 0 ^ p7 = 1 ^ p6 = 0
4. p4 = 0 ^ p1 = 1 ^ p2 = 0

"X" is removed if it satisfies any of the following conditions

1. p4 = 1 ^ p3 = 1
2. p4 = 1 ^ p3 = 0 ^ p5 = 1 ^ A = 1
3. p4 = 0 ^ p1 = 1 ^ p2 = 1
4. p4 = 0 ^ p1 = 0 ^ p7 = 1 ^ p6 = 1 ^ B = 1

The above tests are repeated until no more points can be deleted from the pattern.

RESULT OF TESTING:

CPU TIME .... 20.352 seconds

MEMORY REQUIREMENT .... 26,624 words

CONSERVATION OF TOPOLOGICAL INFORMATION 14

NO. OF PATTERNS TESTED 48

*NOTE:

This algorithm is modified from the program listing supplied by Mr. H. Tamura.

THINNING ALGORITHM B ---

AUTHORS : T. Y. Zhang and C. Y. Suen [22]

TERMINOLOGY:

3 by 3 window

```
+---+---+---+
|P9 |P2 |P3 |
+---+---+---+
|P8 |P1 |P4 |
+---+---+---+
|P7 |P6 |P5 |
+---+---+---+
```

DESCRIPTION OF ALGORITHM:

In the first sub-iteration, the contour point P1 is deleted from the digital pattern if it satisfies all of the following conditions :

(a) $2 <= B(P1) <= 6$

(b) $A(P1) = 1$

(c) $P2 * P4 * P6 = 0$

(d) $P4 * P6 * P8 = 0$

where A(P1) is the number of "01" pairs in ordered set P2, P3, P4, ..., P8, P9 of the neighbours of P1, and B(P1) is the number of nonzero neighbours of P1, that is

$$B(P1) = P2 + P3 + P4 + P5 + P6 + P7 + P8 + P9$$

In the second sub-iteration, a contour point P1 is deleted if the following conditions are satisfied ---

(a) 2 <= B(P1) <= 6

(b) A(P1) = 1

(c') P2 * P4 * P8 = 0

(d') P2 * P6 * P8 = 0

Note: conditions (a) and (b) are unchanged from the first sub-iteration.

The above tests are repeated until no more points can be deleted from the pattern.

RESULT OF TESTING:

CPU TIME .... 24.505 seconds

MEMORY REQUIREMENT .... 26,176 words

CONSERVATION OF TOPOLOGICAL INFORMATION 9

NO. OF PATTERNS TESTED 48

*NOTE:

This algorithm is modified from the program listing supplied by Mr. T.Y. Zhang and Dr. C.Y. Suen.

THINNING ALGORITHM 9 ---

AUTHOR : C. Arcelli [23]

TERMINOLOGY:

3 by 3 window

NW	N	NE
W	P	E
SW	S	SE

DESCRIPTION OF ALGORITHM:

A border point (north, south, east, west) is removed if it has at least two neighbours and satisfies at least one of the following conditions ---

A) $W * \bar{S} * E = 0$ (FOR NORTH BORDER POINT)

B) $E * \bar{N} * W = 0$ (FOR SOUTH BORDER POINT)

C) $N * \bar{W} * S = 0$ (FOR EAST BORDER POINT)

D) $S * \bar{E} * N = 0$ (FOR WEST BORDER POINT)

and also satisfies

$$\bar{W} * NW * \bar{N} + \bar{N} * NE * \bar{E} +$$
$$\bar{E} * SE * \bar{S} + \bar{S} * SW * \bar{W} = 0$$

The above process is repeated until no more changes occur after one complete iteration.

RESULT OF TESTING

CPU TIME .... 24.490 seconds

MEMORY REQUIREMENT .... 26,048 words

CONSERVATION OF TOPOLOGICAL INFORMATION 19

NO. OF PATTERNS TESTED 48

CONCLUDING REMARKS

The resulting skeleton depends heavily on the processes used. The basic step is to scan the matrix and test all the black points and their relations to their neighbouring points. Each author develops his/her own testing criteria to obtain the skeleton of the image. The results reflect the differences and characteristics of each technique developed.

Table I provides a summary of the computation requirement of these algorithms. Tsuruoka's method [20] proved to use the least CPU time while Arcelli's [23] used the least memory space. For parallel operation of large binarized data the skeletonization process requires a large amount of memory which may create a problem, this is particularly accute in Stefanelli and Rosenfeld's algorithm [17].

Apart from the computation requirement of these algorithms, the most important performance lies in the capability of conserving the topological information of the original image. Tables II and III summarize the performance of all the algorithms tested. From this Table, we can see Hilditch's [4] and Ma and Yudin's [16] algorithms preserve very well the topological information contained in the original patterns. On the other hand, Tsuruoka's [20] and Arcelli's [23] give very poor results.

With respect to noise immunity, most of the algorithms show fairly good resistance to noise except Arcelli's algorithm [23] which performs rather poorly in noise. From Table II, we also notice a large number of the available thinning algorithms employ parallel matrix transformation, only Hilditch [4] and H.Ma , J.Yudin [16] use sequential operators. When choosing a thinning algorithm, one has to consider the computational requirement as well as the performance of each algorithm. This study provides a cross comparison among these algorithms which might be used as a guide for choosing thinning algorithms.

TABLE I

COMPUTATION REQUIREMENT

ALGORITHMS	CPU TIME (SECS)	MEMORY (WORDS)
1. E. Deutsch	27.896	26,240
2. C. J. Hilditch	20.401	26,496
3. H. Ma & J. Yudin	23.438	26,752
4. R. Stefanelli & A. Rosenfeld	49.653	29,696
5. D. Rutovitz	30.005	28,096
6. H. Tamura	26.269	26,688
7. S. Tsuruoka	20.352	26,624
8. T. Y. Zhang & C. Y. Suen	24.505	26,176
9. C. Arcelli	24.490	26,048

TABLE II

CHARACTERISTICS

ALGORITHMS	CONNECTIVITY		OPERATOR	
	4-WAY	8-WAY	SEQUENTIAL	PARALLEL
1. E. Deutsch		X		X
2. C. J. Hilditch		X	X	
3. H. Ma & J. Yudin		X	X	
4. R. Stefanelli & A. Rosenfeld		X		X
5. D. Rutovitz	X			X
6. H. Tamura	X	X		X
7. S. Tsuruoka	X	X		X
8. T. Y. Zhang & C. Y. Suen	X	X		X
9. C. Arcelli	X	X		X

TABLE III

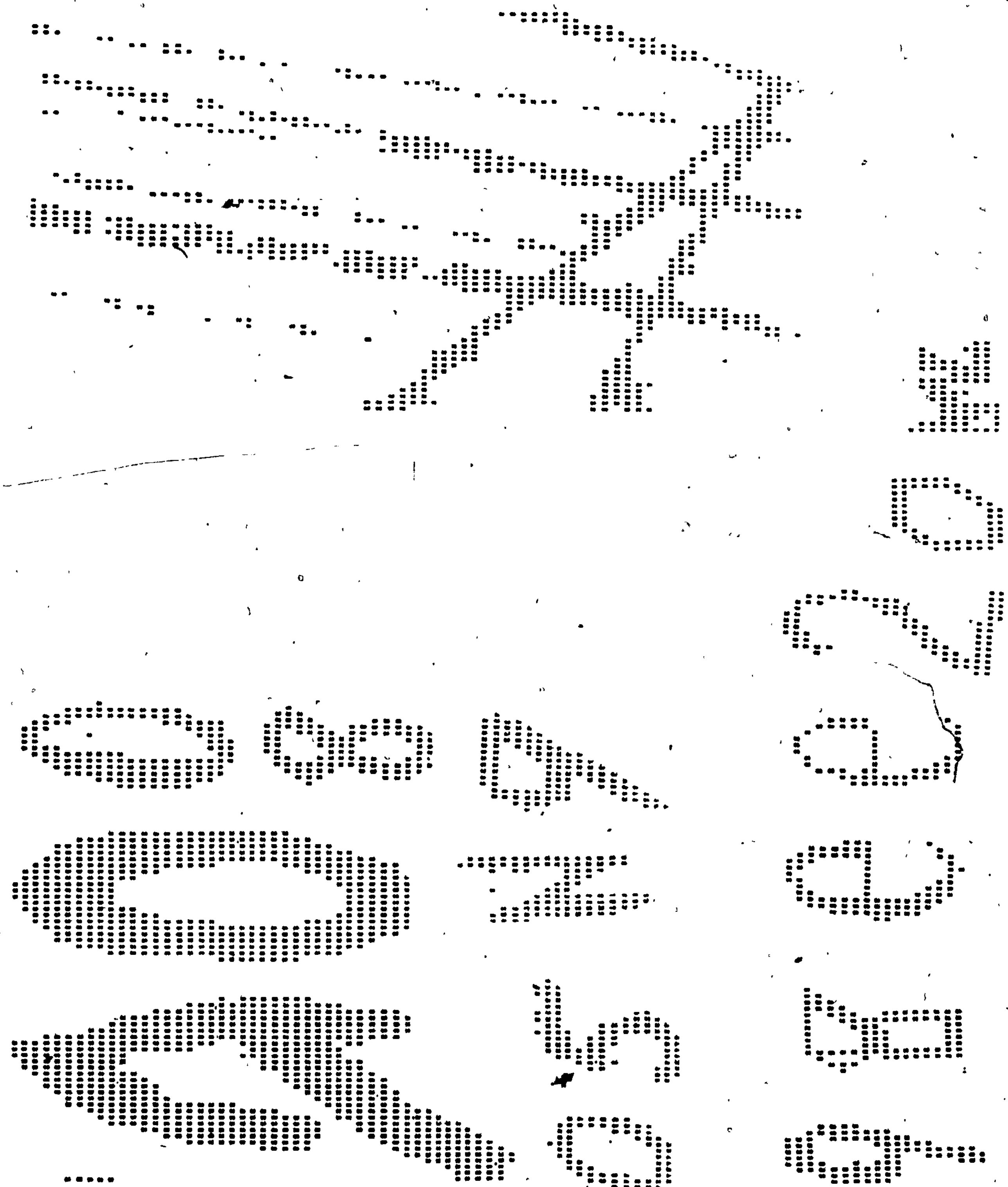
CONSERVATION OF TOPOLOGICAL INFORMATION

ALGORITHMS	CONNEC-TIVITY	EXCESSIVE EROSION OR LOSS OF LIMBS	NOISE IMMUNITY	TOTAL
1. E. Deutsch	1	2	2	5
2. C. J. Hilditch	0	0	2	2
3. H. Ma & J. Yudin	0	0	0	0
4. R. Stefanelli & A. Rosenfeld	7	3	3	13
5. D. Rutovitz	0	2	2	4
6. H. Tamura	1	5	3	9
7. S. Tsuruoka	6	0	8	14
8. T. Y. Zhang & C. Y. Suen	0	2	7	9
9. C. Arcelli	0	5	14	19

Conservation of topological information is expressed in number of skeletonized patterns visually inspected to have problems. Connectivity is determined by the continuity of the medial line, that is, let p be a set of points of the skeleton, p = p0,p1,p2,...pn such that pi is a neighbour of pi-1 where 0<=i<=n. Excessive erosion is the loss of end points or limbs from line-like patterns, for example, the "arm" has been amputated in the skeleton (see example ED, p.40 in Appendix). Noise resistance is based on the sensitivity of each algorithm to conditions such as uneven edges or holes in the original pattern. The resulting skeletons affected by noise immunity in case of very "thick" original pattern often have excessive branches (see example ST, p.39 in Appendix). The above table gives the statistical results of thinning all the 48 test patterns. The total is the sum of connectivity, excessive erosion and noise immunity.

APPENDICES

Appendix A - All 48 tested patterns.



Appendix B – Five examples using the nine algorithms.

Notations:

ED – E.S. Deutsch

CH – C.J. Hilditch

MY – H. Ma and J. Yudin

SR – R. Stefanelli and A. Rosenfeld

DR – D. Rutovitz

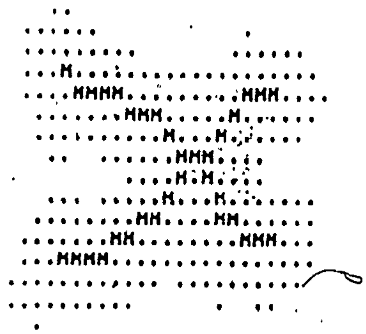
HT – H. Tamura

ST – S. Tsuruoka

ZS – T.Y. Zhang and C.Y. Suen

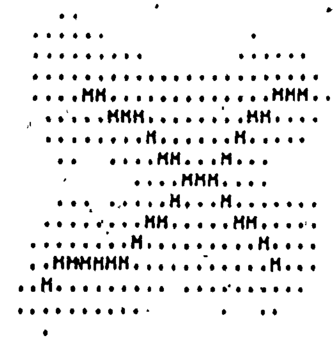
CA – C. Arcelli

EU



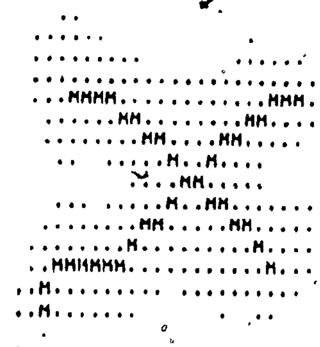
EXECUTION TIME = .20200 SECS

CH



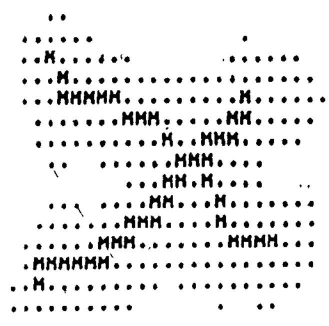
EXECUTION TIME = .12500 SECS

MY



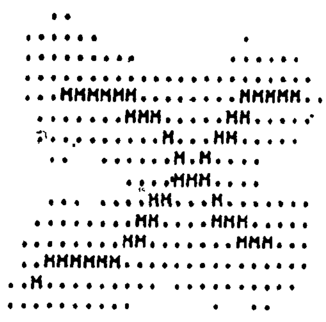
EXECUTION TIME = .20000 SECS

SR



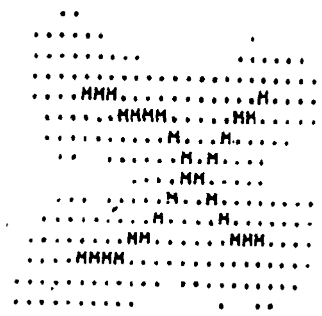
EXECUTION TIME = .29600 SECS

DR



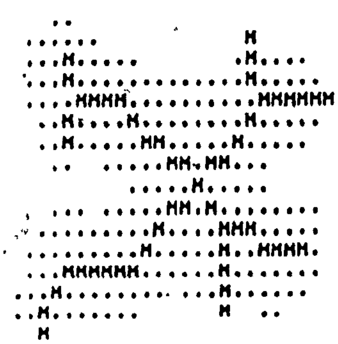
EXECUTION TIME = .49500 SECS

HT



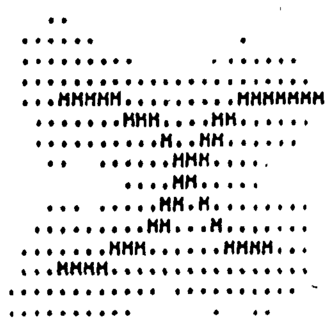
EXECUTION TIME = .34700 SECS

ST



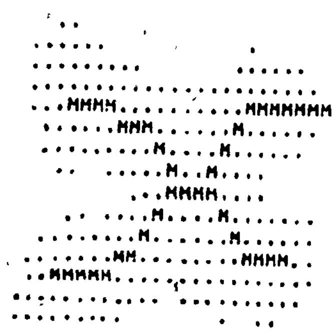
EXECUTION TIME = .10500 SECS

ZS



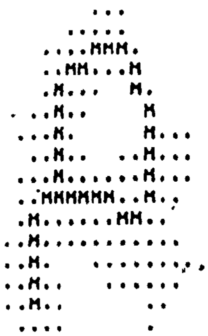
EXECUTION TIME = .16800 SECS

CA



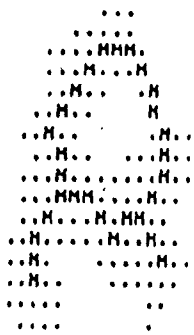
EXECUTION TIME = .35800 SECS

ED



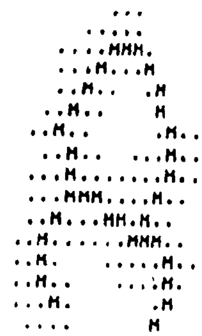
EXECUTION TIME = ..08200 SECS

CH



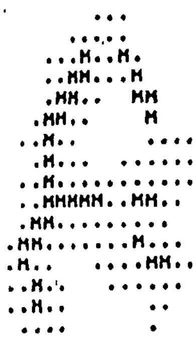
EXECUTION TIME = .05200 SECS

MY



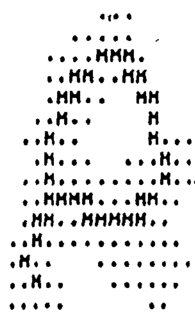
EXECUTION TIME .10900 SEC

SR



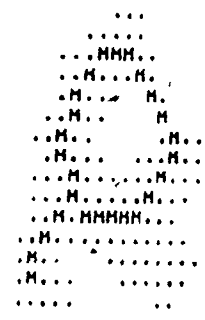
EXECUTION TIME = .13200 SECS

DR



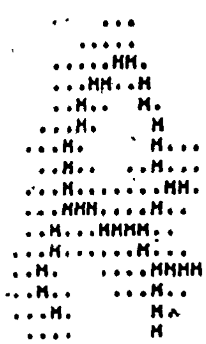
EXECUTION TIME = .35000 SECS

HT



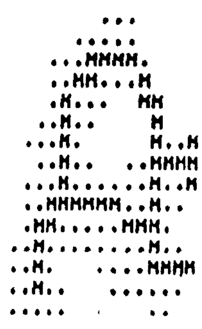
EXECUTION TIME = .11300 SECS

ST



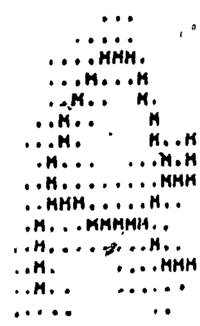
EXECUTION TIME = .04900 SECS

ZS



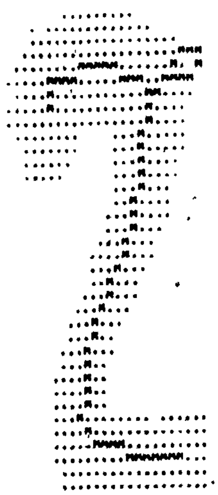
EXECUTION TIME = .06600 SECS

CA



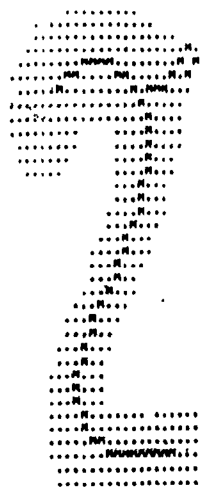
EXECUTION TIME = .14900 SECS

FD



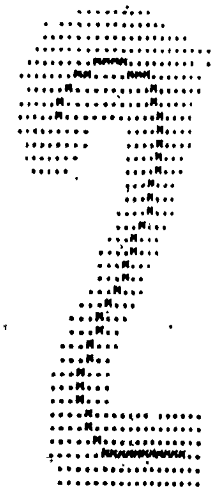
EXECUTION TIME = .32900 SECS

CH



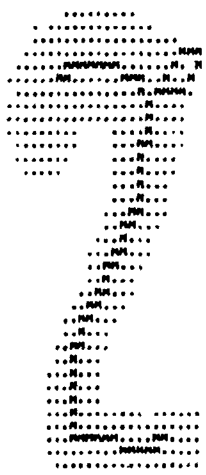
EXECUTION TIME = .22900 SECS

MY



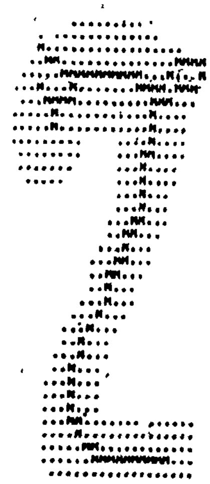
EXECUTION TIME = .41100 SECS

SR



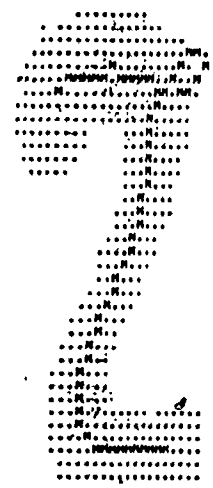
EXECUTION TIME = .37000 SECS

DR



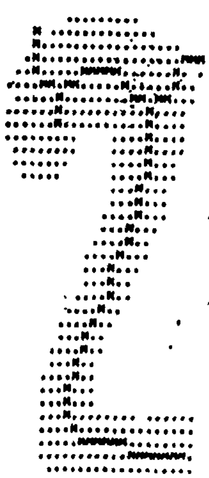
EXECUTION TIME = .71700 SECS

WT



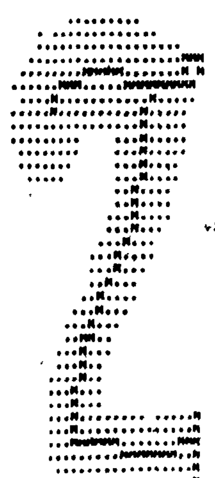
EXECUTION TIME = .39500 SECS

ST



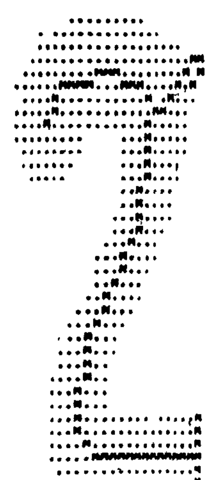
EXECUTION TIME = .28000 SECS

ZS



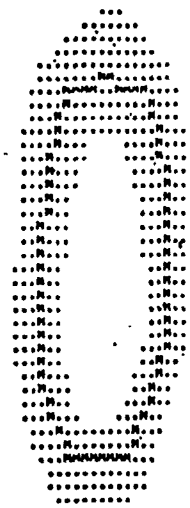
EXECUTION TIME = .29000 SECS

CA



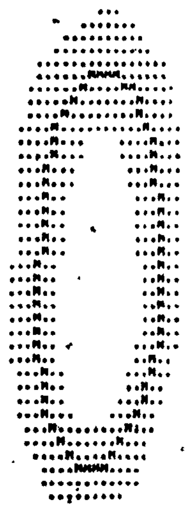
EXECUTION TIME = .30900 SECS

ED



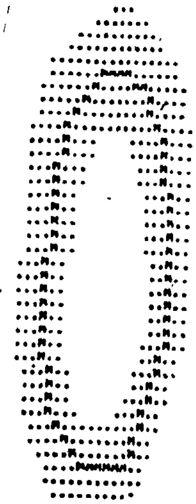
EXECUTION TIME = .20700 SECS

CM



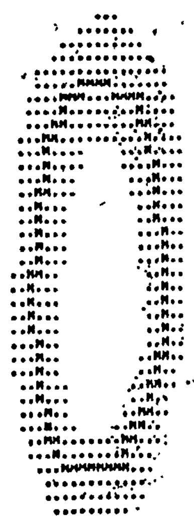
EXECUTION TIME = .21100 SECS

MY



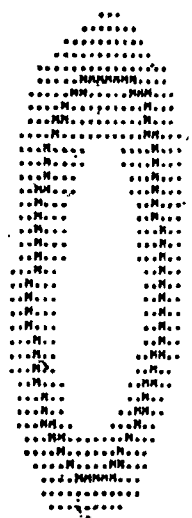
EXECUTION TIME = .34900 SECS

BR



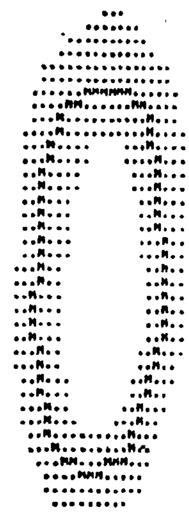
EXECUTION TIME = .43200 SECS

DR



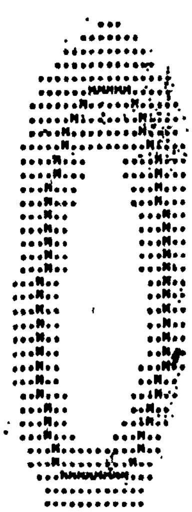
EXECUTION TIME = .70200 SECS

MT



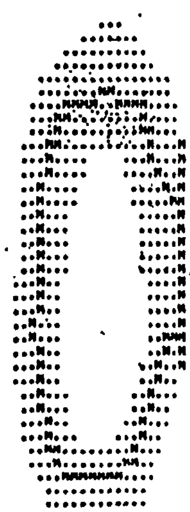
EXECUTION TIME = .52200 SECS

BT



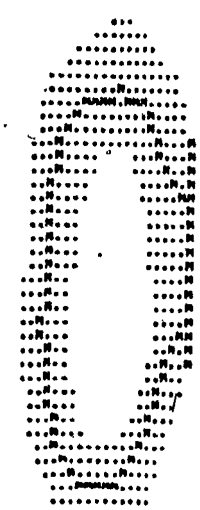
EXECUTION TIME = .17100 SECS

ZO



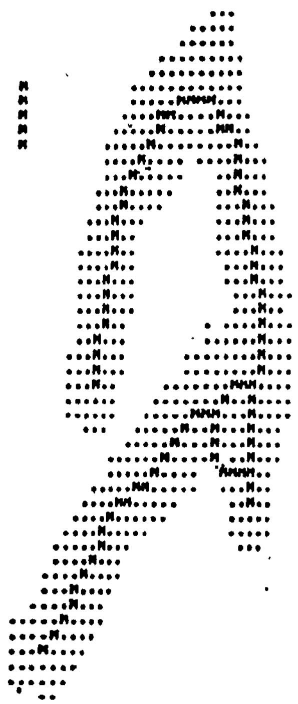
EXECUTION TIME = .39000 SECS

CA



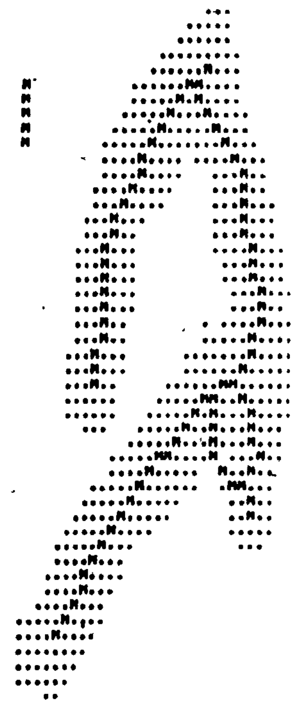
EXECUTION TIME = .63300 SECS

ED.



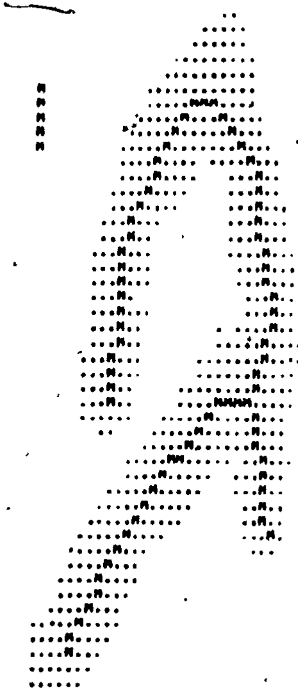
EXECUTION TIME = .42300 SECS

CH



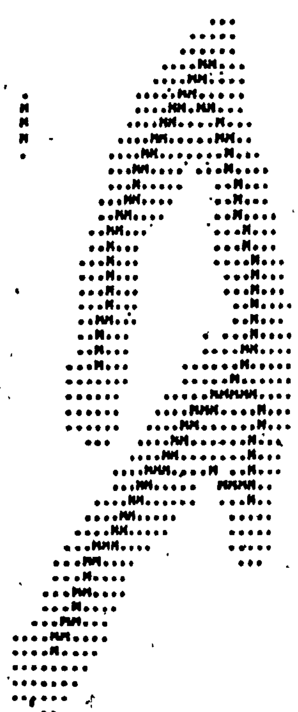
EXECUTION TIME = .32200 SECS

MY



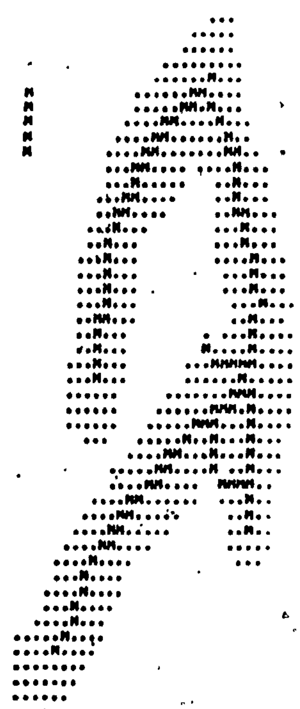
EXECUTION TIME = .63500 SECS

SR



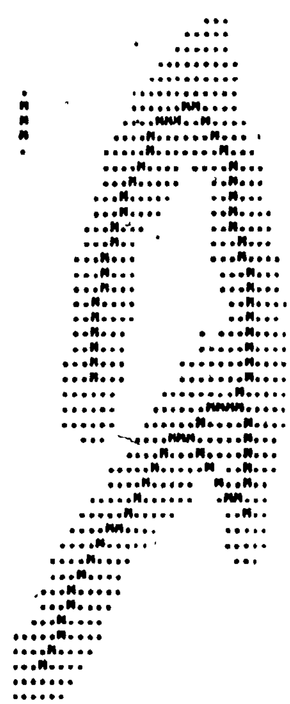
EXECUTION TIME = .74300 SECS

BR



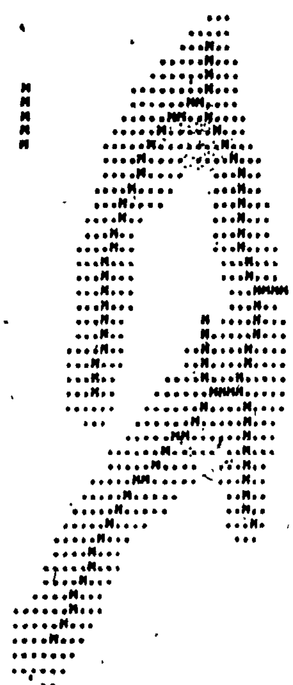
EXECUTION TIME = 1.24600 SECS

HT



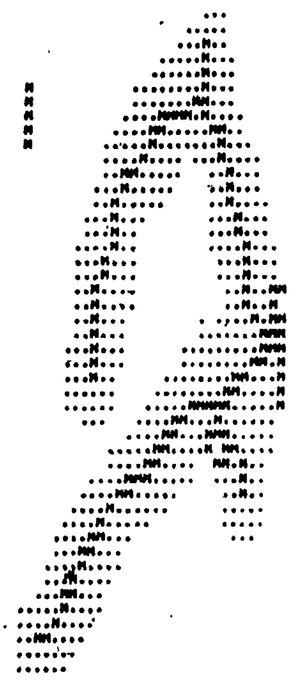
EXECUTION TIME = .71600 SECS

ST



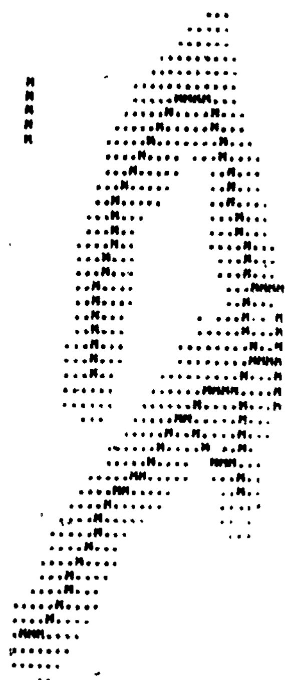
EXECUTION TIME = .28400 SECS

ZS



EXECUTION TIME = .42600 SECS

CA



EXECUTION TIME = .02600 SECS

BIBLIOGRAPHY

1. H. Sherman, "A quasi-topological method for the recognition of line patterns", Information Processing, Proc. UNESCO conf. pp. 232-238, Butterworths, London, 1959.

2. A. P. Pullen, "Automatic visual inspection of complex industrial components", paper read at BPRA meeting in University College, London, February 1977.

3. R. N. Dixon and C. J. Taylor, "Automated asbestos fibre counting", Proc. IOP Conf. on Machine-aided Image Analysis, pp. 178-185. Institute of Physics, London 1979.

4. C. J. Hilditch, "Linear skeletons from square cupboards", in Machine Intelligence IV, B. Meltzer and D. Michie(eds.), pp. 403-420. Edinburgh University Press, Edinburgh 1969.

5. J. F. O'Callaghan and J. Loveday, "Quantitative measurements of soil cracking patterns", Pattern Recognition, Vol. 5, pp. 83-98, 1973.

6. C. V. K. Rao, B. Prasada and K. R. Sarma, "An automatic fingerprint classification system", Proc. 2nd Int. Joint Conf. on Pattern Recognition, pp. 180-184, 1974.

7. B. Moayor and K. S. Fu, "A tree system approach for fingerprint pattern recognition", IEEE trans. Comput., Vol. C-25, pp. 262-274, 1976.

8. M. Beun, "A flexible method for automatic reading of handwritten numerals", Philips Tech. Rev., Vol. 33, pp. 89-101 and vol. 33, pp. 130-137, 1973.

9. K. J. Udupa and I. S. N. Murthy, "Some new concepts for encoding line patterns", Pattern Recognition, Vol. 7, pp. 225-233, 1975.

10. I. D. Judd, "Compression of binary images by stroke encoding", Comput. Digital Techniques, Vol.2, pp.41-48, 1979.

11. G. F. P. Deeker and J. P. Penny, "On interactive map storage and retrieval", Information, Vol. 10, pp. 62-74, 1972.

12. T. Kreifelts, "Skelettierung und Linienverfolgung in Raster digitalisierten linienstrukturen", GL/NTG Congress on Digital Image Processing, Munich, March 1977. Springer, Informatik- Fachberichte, Nr 8.

13. P. Seuffert, "An application of line and character recognition in cartography", Proc. IEEE Congress on Pattern Recognition, pp. 338-343. Troy, New York, July 1977.

14. H. Freeman, "On the encoding of arbitrary geometric configurations", IRE Trans. Electron. Comput., Vol. EC-10, pp. 260-268, 1961.

15. E. S. Deutsch, "Thinning algorithms on rectangular, hexagonal, and triangular arrays", Communications of ACM, Volume 15, No. 9, pp. 827- 837, September 1972.

16. Helen Ma and Janice Yudin, Skeletonization, Term Report, Dept. of Computer Science, Concordia University, December 1979.

17. R. Stefanelli and A. Rosenfeld, "Some parallel thinning algorithms for digital pictures", Journal of the Association for Computing Machinery, Vol.18, No. 2, pp. 255 - 264, April 1971.

18. Denis Rutovitz, "Pattern recognition", J. Royal Statists. Soc., Vol. 129, Series a, pp. 504 - 530, 1966.

19. Hideyuki Tamura, "A comparison of line thinning algorithms from digital geometry viewpoint", Proc. 4th Int. Joint Conf. on Pattern Recognition, pp. 715 - 719, 1978.

20. S. Tsuruoka - From H. Tamura

21. Shigeki Yokoi and et al., "Topological Properties in Digitized Binary Pictures", Systems- Computers- Controls, Vol. 4, No. 6, 1973 Translated from Denshi Tsushin Gakkai Ronbunshi, Vol. 56-D, No. 11, pp. 662 - 669, November 1973.

22. T. Y. Zhang and C. Y. Suen, "A parallel algorithm for thinning digital patterns", in press, Communications of the ACM.

23. C. Arcelli, "A condition for digital points removal", Signal Processing, vol 1, pp. 283 - 285, 1979.

24. T. Pavlidis, "A flexible parallel thinning algorithm ", Proc. IEEE Computer Society Conf. Pattern Recognition and Image Processing, pp. 162 - 167, August 1981.