Pattern Recognition Letters 11 (1990) 25–33
North-Holland

January 1990

Parametrisable skeletonization of binary and multi-level images

S. RIAZANOFF

CISI Ingénierie, DGLA, 3 rue Le Corbusier, SILIC 232, 94528 Rungis Cedex, France, and Laboratoire de Minéralogie-Cristallographie (CNRS), Tour 16, Universités P. et M. Curie et Paris 7, 4 place Jussieu, 75252 Paris Cedex 5, France

B. CERVELLE

Laboratoire de Minéralogie-Cristallographie (CNRS), Tour 16, Universités P. et M. Curie et Paris 7, 4 place Jussieu, 75252 Paris Cedex 5, France

J. CHOROWICZ

Laboratoire de Géologie Structurale (CNRS), Tour 26, Université P. et M. Curie, 4 place Jussieu, 75252 Paris Cedex 5, France

Received 15 March 1989

Abstract: In this paper, a new definition of the skeleton of an object, called the *curvature skeleton*, is presented. This definition is based on the curvature maxima computed along the object's contours. Some principles of this mathematical model are used to implement a new method of parametrisable skeletonization of binary images. The binary algorithm also fits multi-level (gray scale) image skeletonization.

Key words: Skeleton, contour processing algorithm, thinning algorithm, ridges extraction.

1. Introduction

Skeletonization, more widely known as pattern thinning, is a major issue in image processing. Among various works concerning this subject, one may distinguish two classes of methods. The more usual algorithms, based on Hilditch's views (1969), process several passes (iterations) on a binary image. For each pass, the neighbourhood configuration (usually 3∗3) of each pixel has to be tested. According to this test, the pixel is either removed or kept.

More recently, authors such as Martinez-Perez (1987) or Shapiro (1981) have proposed algorithms based on the object's contour processing in images either in raster format or already in vector form. The latter are generally faster, but they may not be always suitable for complex images because they sometimes cut off the skeleton.

Nevertheless, the object's contour processing seems to be a very interesting approach. We propose a new mathematical modeling (Section 2) showing the local maxima of the curvature, computed along the object's contour, which allows the generation of the skeleton branches.

A new definition of skeleton, called the *curvature skeleton*, extends beyond the usual definition quoted by Shapiro (1981): "The skeleton (sometimes called 'medial axis' or 'symmetric axis') of the inside of a black object or 'blob' on a white background in a two-dimensional Euclidian plane is the set of points (x) such that there are at least two points on the object's contour that are equidistant from (x) and are closest to (x)."

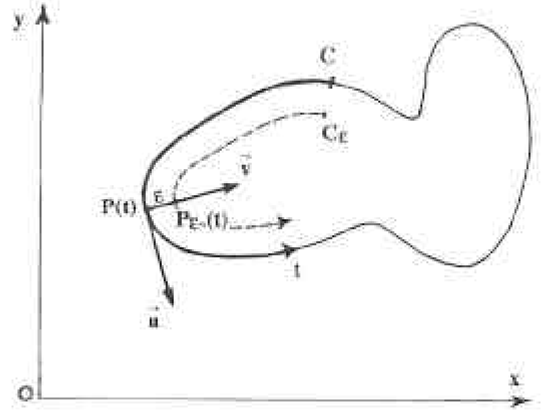

Figure 1. Generation of the *translated curve* C_ε from the object's contour C.

Some aspects of our new mathematical formalism are parts of the binary skeletonization algorithm. The aim of our method is to obtain, choosing the right parameters, the skeleton which fits perfectly the user's needs.

Our process, close to Hilditch's approach (1969), is relatively expensive in terms of computing time. However, it is always possible to reduce this time, for example by using a contour processing method as described by Van Vliet and Verwer (1988).

The principles used in binary images skeletonization are reintroduced in multi-level (gray scale) image processing. With a didactic view in mind, we have made a multi-level test image from the binary image showing the letter 'A'. The results obtained for each of these images allow the reader to estimate the accuracy of the relation between binary and multi-level processing.

2. Mathematical model of the curvature skeleton

Notations

Let E be an object and C its contour. C is a closed curve or a set of closed curves when E has holes. It is assumed in this section that each closed curve is at least twice derivable in each point.

C is denoted in parametrical form:

$$C: \left\{ P(t) = \begin{pmatrix} x(t) \\ y(t) \end{pmatrix} \middle| \; t = 0, \ldots, T \right\},$$

with t varying trigonometrically.

The curvature $K(t)$ in point $P(t)$ is given by:

$$K(t) = \frac{\begin{vmatrix} x' & y' \\ x'' & y'' \end{vmatrix}}{(x'^2 + y'^2)^{3/2}},$$

$$K(t) = \frac{x'y'' - x''y'}{(x'^2 + y'^2)^{3/2}}. \tag{1}$$

Erosion of E by a disk of radius ε (denoted E_ε), is defined as the set of points belonging to E and at a distance from the contour C greater than or equal to ε.

Let

$$\boldsymbol{u} = \begin{pmatrix} x' = \dfrac{\mathrm{d}(x(t))}{\mathrm{d}t} \\ y' = \dfrac{\mathrm{d}(y(t))}{\mathrm{d}t} \end{pmatrix}$$

be the vector tangent to C in $P(t)$, and let

$$\boldsymbol{v} = \begin{pmatrix} -y' \\ x' \end{pmatrix}$$

be the vector normal to C in $P(t)$.

C_ε is defined as the point by point *translated curve* from C, with a distance ε along the $\boldsymbol{v}$ vector (Figure 1):

$$P_\varepsilon(t) = \begin{cases} x_\varepsilon(t) = x(t) - \varepsilon \dfrac{y'}{\sqrt{x'^2 + y'^2}}, \\ y_\varepsilon(t) = y(t) + \varepsilon \dfrac{x'}{\sqrt{x'^2 + y'^2}}. \end{cases} \tag{2}$$

Definition of the curvature skeleton

$P_\varepsilon(t)$ belongs to the skeleton if and only if:

(1) $P_\varepsilon(t)$ belongs to the erosion E_ε, and

(2) D1 or D2 or D3 holds:

D1. $P_\varepsilon(t)$ is a curvature local maximum along C_ε.

D2. x and y derivatives are zero in $P_\varepsilon(t)$.

D3. The C_ε curve intersects itself in $P_\varepsilon(t)$, i.e. $\exists t' \neq t$, with $P(t') = P(t)$.

The full curvature skeleton is achieved when ε takes all values from 0 to infinity (or at least while the $E\varepsilon$ erosion is not empty).

The three cases D1, D2 and D3 are dependent on the value ε in relation to the curvature radii $1/K(t)$ computed within one interval of the original curve C:

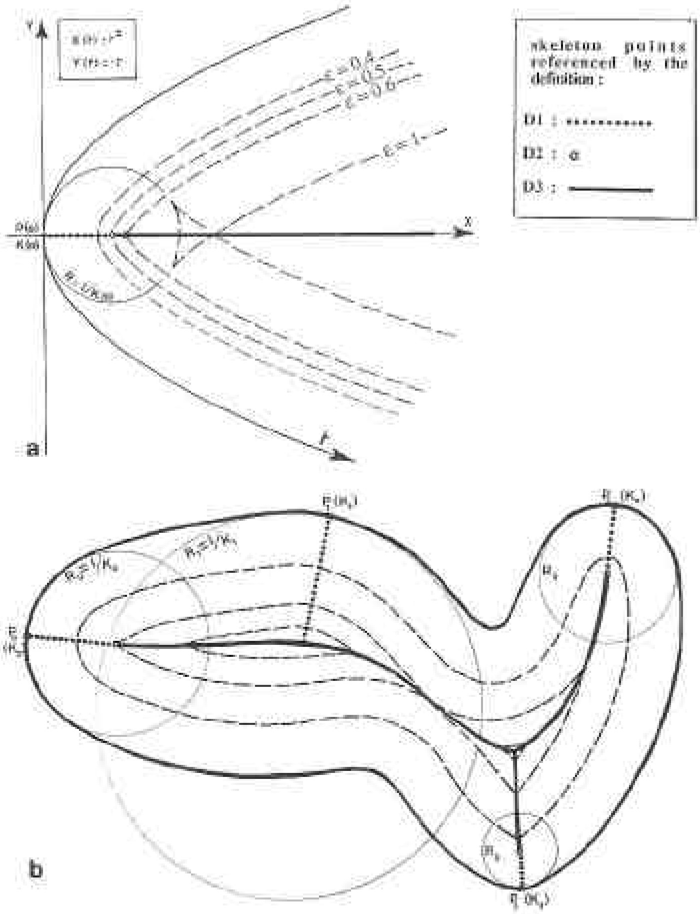

Figure 2. *Curvature skeleton* achieved by applying definitions D1, D2 and D3. Circles show the curvature radii R in points P_i with maximal curvature K along the original curve C. (a) Simple case showing the *translated curve* C_ε for several values of ε. (b) An ordinary case showing only the erosion contours E_ε.

D1: case of $\varepsilon < 1/K(t)$ for all points of the interval,

D2: case of $\varepsilon = 1/K(t)$ for only one point of the interval,

D3: case of $\varepsilon > 1/K(t)$ for at least one point of the interval,

and are shown in Figures 2a and 2b.

For each curvature maximum value K_{max} along the original curve C, we drew the circle located at the curvature center whose radius value is $1/K_{max}$.

Figure 2a depicts a theoretical situation of a curve whose parametrical equation is locally:

$$C: \begin{pmatrix} x(t) = t^2 \\ y(t) = -t \end{pmatrix}.$$

Figure 2b shows some ordinary object whose curvature skeleton was drawn by hand.

In Figure 2a, when ε is greater than 1/2, the C_ε curve produces one 'loop' and two 'return-points' within the neighbourhood of $P_\varepsilon(0)$. In Figure 2b, these 'loops' have been removed to keep only the E_ε erosion contours.

Discussion

Skeleton points referenced by the D3 definition are exactly those referenced by the classical definition quoted by Shapiro (1981).

Definition D1 allows us to extend the usual skeleton to reach the object's contour.

Definition D2 references the points joining the

D1 definition points and the D3 definition points.

Curvature properties of the translated curve

In this section we search for one relation between the curvature in a point $P(t)$ and the curvature in its *translated point* $P_\varepsilon(t)$.

It is easy to prove that the translated curve C_ε contains the contour of the E_ε erosion.

Using equations (2) and (1), the first derivative in $P_\varepsilon(t)$ is:

$$\boldsymbol{u}_\varepsilon = \begin{cases} x'_\varepsilon = x'(1-\varepsilon K), \\ y'_\varepsilon = y'(1-\varepsilon K), \end{cases} \tag{3}$$

where K is the curvature in $K(t)$ of the original curve C. We note that $\boldsymbol{u}_\varepsilon$ and $\boldsymbol{u}$ are co-linear (when $K \neq 1/\varepsilon$). This fact allows us to prove that the $(\varepsilon + \varepsilon')$ radius erosion $E_{\varepsilon+\varepsilon'}$ is equal to the ε radius erosion of the ε' radius erosion:

$$E_{\varepsilon+\varepsilon'} = (E_{\varepsilon'})_\varepsilon = (E_\varepsilon)_{\varepsilon'}.$$

Coordinates of the second derivative are:

$$\frac{\mathrm{d}^2 OP(t)}{\mathrm{d}t^2} = \begin{cases} x''_\varepsilon = x''(1-\varepsilon K) - \varepsilon x'K', \\ y''_\varepsilon = y''(1-\varepsilon K) - \varepsilon y'K'. \end{cases} \tag{4}$$

The curvature $K_\varepsilon(t)$ in the point $P_\varepsilon(t)$ is given by:

$$K_\varepsilon(t) = \frac{\begin{vmatrix} x'_\varepsilon & y'_\varepsilon \\ x''_\varepsilon & y''_\varepsilon \end{vmatrix}}{(x'^2_\varepsilon + y'^2_\varepsilon)^{3/2}},$$

and using equations (3), (4) and (1):

$$K_\varepsilon(t) = \frac{K}{|1-\varepsilon K|}. \tag{5}$$

In each point, when $\varepsilon < 1/K(t)$, we notice that:

– convexity: $K_\varepsilon > K > 0$, translated curves are more and more convex,

– concavity: $K < K_\varepsilon < 0$, translated curves are less and less concave.

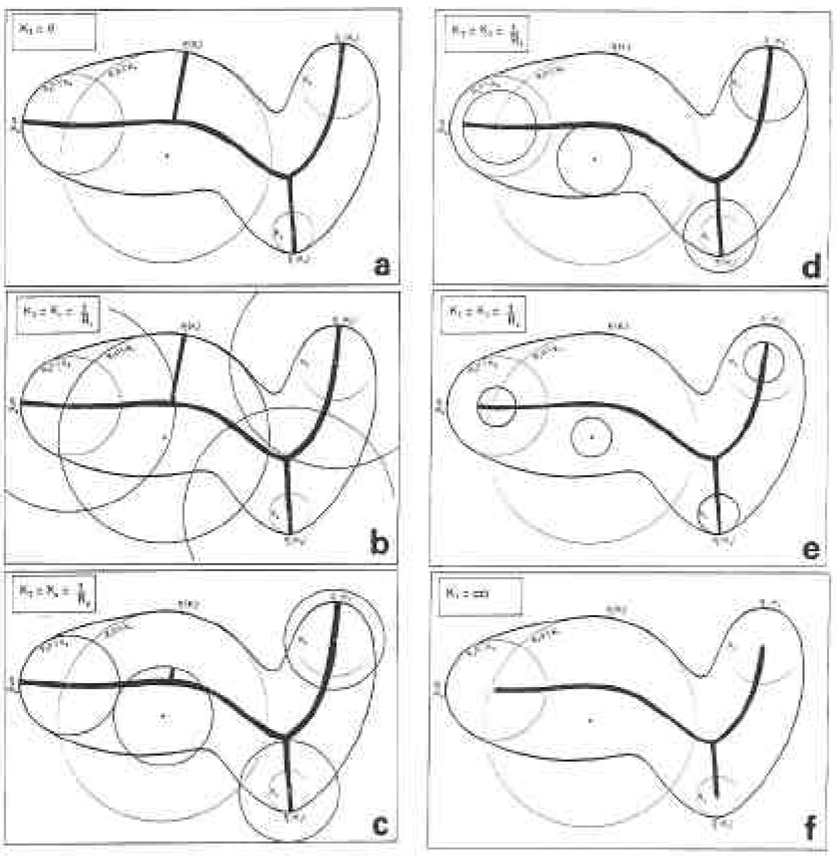

Figure 3. *Curvature skeleton* of the object (Figure 2a) for several threshold values.

Curvature variations are given by:

$$K'_\varepsilon(t) = \frac{\mathrm{d}(K_\varepsilon(t))}{\mathrm{d}t} = \frac{K'}{|1-\varepsilon K|(1-\varepsilon K)}. \tag{6}$$

When ε is less than $1/K(t)$, K_ε' and K' have the same sign and take the value 0 simultaneously.

In particular, if $P(t)$ is a curvature maximum ($K' = 0$), then $P_\varepsilon(t)$ is a curvature maximum ($K_\varepsilon' = 0$). This proves some *spatial continuity* (contiguity) of the curvature skeleton points referenced by the D1 definition.

Thresholding of the curvature skeleton

Let $P_\varepsilon(t)$ be the skeleton points applying our definitions. On this set, we define the *curvature function* as:

$$\mathscr{K}(P_\varepsilon(t)) = \begin{cases} K_\varepsilon(t), & \text{in case D1}, \\ +\infty, & \text{in case D2 or D3}. \end{cases}$$

Removing from the image the skeleton points whose curvature $\mathscr{K}(P_\varepsilon(t))$ is less than a given threshold K_T, we obtain a new skeleton, which is always 'connected', albeit with a reduced size (Figure 3).

In these figures, the disk centered at the curvature center whose radius is $1/K_T$ has been added in bold print. Points referenced by the D1 definition and located without the disk have been removed.

Figure 3a shows a threshold equal to 0; the full skeleton resulting from definitions D1, D2 and D3 is unchanged (see Figure 2b).

Figure 3f shows an infinite threshold; only points referenced by the definitions D2 and D3 are saved.

Figures 3b, 3c, 3e and 3f describe intermediate situations corresponding respectively to the increasing threshold values K_1, K_0, K_2 and K_3, these last values being the curvature local maxima computed along the original curve C (Figure 2b).

These results may be considered as an answer to the 'extremity points' processing problem which is frequently encountered in the literature. For instance, Pavlidis (1980) uses a particular processing with these points.

The thresholding allows us to retain among the skeleton branches only those (or parts of them) generated from points whose convexity is sharp enough.

Conclusions

The curvature skeleton definition extends and completes the usual one. The curvature local maxima along the original curve are the starting points of branches which converge on the central part of the skeleton.

The possible noise may be cleared by the curvature thresholding method.

Lastly, the curvature skeleton is larger than the usual skeleton. The size of the original object is recalled in a more exacting way, which may be important in case of subsequent measurements.

3. Binary skeletonization

Without being a rigorous application, the binary skeletonization algorithm described here is largely inspired by the preceding.

Unlike searching for the curvature local maxima, we allow the skeleton to *start* only from points whose convexity is greater than a given threshold K_T.

Notations

Let a binary image be given. We assign to object pixels value one and to background pixels value zero. The operations treated in this paper are based on 3∗3 neighbourhoods:

$b3$ $b2$ $b1$
$b4$ a $b0$ (setting $b0 = b8$)
$b5$ $b6$ $b7$

Groen and Foster (1984) define:

the number of 4-connected neighbours:

$$\varphi_4 = \sum_{k=0,2,4,6} b_k,$$

the number of 8-connected neighbours:

$$\varphi_8 = \sum_{k=0,\ldots,7} b_k.$$

We redefine the Hilditch crossing number in 4-connected neighbourhood:

$$\gamma_4 = \sum_{k=0,2,4,6} h_k \quad \text{with } h_k = 1$$

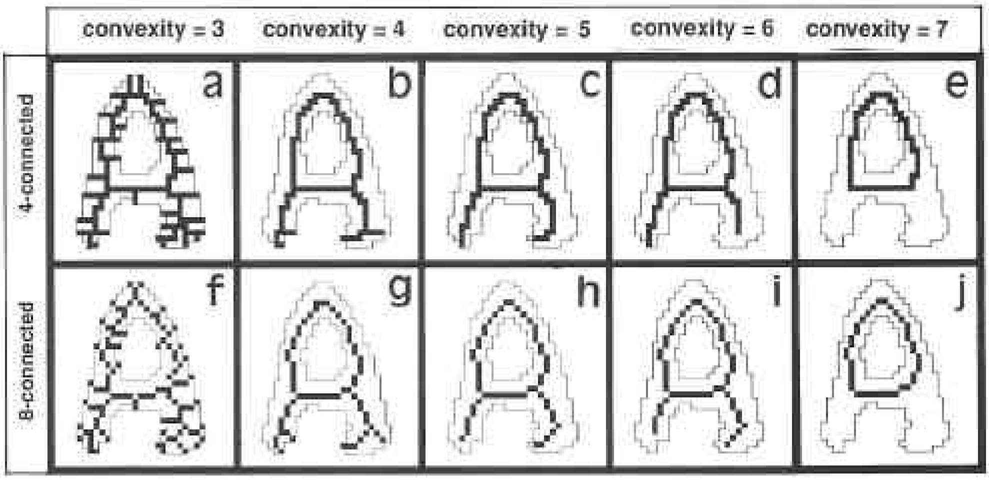

Figure 4. Binary skeletons achieved in 4-connected neighbourhood (a–e) and in 8-connected neighbourhood (f–j) and for several values of the threshold $K_T = 3,4,5,6$ and 7.

$$\text{iff } (b_k = 1) \text{ and } ((b_{k+1} = 0) \text{ or } (b_{k+2} = 0)),$$

in 8-connected neighbourhood:

$$\gamma_8 = \sum_{k=0,2,4,6} h_k \quad \text{with } h_k = 1$$

$$\text{iff } (b_k = 0) \text{ and } ((b_{k+1} = 1) \text{ or } (b_{k+2} = 1)).$$

Object pixels which have at least one of the 4 neighbours belonging to the background ($a = 1$ and $\varphi_4 < 4$) are called *contour pixels*. In each iteration, only these may be removed.

Pixels which link at least two parts of the object ($\gamma_4 > 1$ in 4-connected neighbourhood or $\gamma_8 > 1$ in 8-connected neighbourhood) are called *connection pixels*. In all skeletonization processes, this kind of points cannot be removed.

The convexity of each contour pixel is roughly estimated by the maximum number of contiguous background pixels among the 8 neighbours.

Examples:

```
*  *  *     *  o  o     o  *  o
o  *  *     o  *  o     o  *  *
o  o  o     o  o  o     o  o  *
 K = 4       K = 7       K = 4
```

Algorithm

Two images are processed by the algorithm. One corresponds to the result in the preceding iteration (Ima(i,j)), the other to the current image (Ima$'(i,j)$).

4-connected skeletonization

1. Preceding and current images are initialized to original image values
2. Repeat
   3. For each pixel (i,j) do
      4. If the pixel belongs to the object (Ima$(i,j) = 1$)
         5. If the pixel belongs to the contour ($0 < \varphi_4 < 4$)
            6. If the convexity is *less than* the threshold ($K < K_T$)
               7. If the pixel is not a *connection point* in preceding image ($\gamma_4 \leqslant 1$)
                  8. If the pixel is not a *connection point* in current image ($\gamma'_4 \leqslant 1$)
                     9. Then remove this pixel from current image (Ima$'(i,j) \leftarrow 0$)
                  endif
               endif
            endif
         endif
      endif
   enddo
   10. Current image becomes preceding image (Ima $\leftarrow$ Ima$'$)

Until any one pixel may be removed from the image.

The 8-connected algorithm is achieved by replacing the γ_4 and γ'_4 tests of statements 7 and 8 by γ_8 and γ'_8 tests.

The no-disconnection test in the preceding image (statement 7) allows us not to erode the objects more strongly across scanning.

The no-disconnection test in the current image (statement 8) ensures not to cut off objects 2 pixels wide.

Results

Figure 4 shows several skeletons obtained from a binary image (32*32) depicting the letter 'A'. Processes were performed in 4-connected neighbourhood (Figures 4a to 4e), in 8-connected neighbourhood (Figures 4f to 4j) and with several convexity threshold values ($K_T = 3,4,5,6$ and 7).

The lowest threshold $K_T = 3$ produces branches from each irregularity of the contour.

The highest threshold $K_T = 7$ involves the total erosion of branches which have free ends. Only circular structures may be saved.

According to the authors, skeletons usually encountered in literature look like those obtained for threshold values 4, 5 or 6.

4. Multi-level skeletonization

In this section, a multi-level (gray scale) image will be treated like a relief in which each pixel is representing one altitude value.

Notations

Our method follows from binary skeletonization. Notions are substituted in this way:

binary image	→	multi-level image,
neighbour belonging to background	→	neighbour located lower than center,
neighbour belonging to object	→	neighbour located higher than center,
connection point	→	col (or saddle point),
skeleton	→	ridge line.

Nevertheless, the object and background notions (and thus the object contour notion) are meaning-

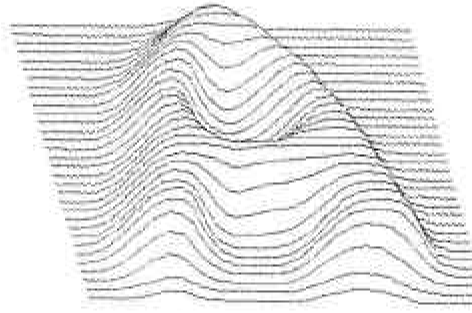

Figure 5. Multi-level image resulting from smoothings in the binary image showing the letter 'A'.

less in terms of multi-level images. Because of this important difference, the algorithm has to be modified: whereas, in binary images, only contour pixels are processed, in multi-level images, all image pixels have to be processed in each iteration.

We define the *center difference* function Δ which transforms the 8 neighbours $b_0, \ldots, b_7$ into the values $d_0, \ldots, d_7$:

$$\forall i = 0, \ldots, 7 \quad \Delta(b_i) = d_i = \begin{cases} +1 & \text{if } b_i \geqslant a, \\ 0 & \text{otherwise.} \end{cases}$$

To these new neighbours (d_i), the definitions of the last section are applied:

$$\varphi_4 = \sum_{k=0,2,4,6} d_k,$$

$$\varphi_8 = \sum_{k=0,\ldots,7} d_k,$$

$$\gamma_4 = \sum_{k=0,2,4,6} h_k \quad \text{with } h_k = 1$$
$$\text{iff } (d_k = 1) \text{ and } ((d_{k+1} = 0) \text{ or } (d_{k+2} = 0)),$$

$$\gamma_8 = \sum_{k=0,2,4,6} h_k \quad \text{with } h_k = 1$$
$$\text{iff } (d_k = 0) \text{ and } ((d_{k+1} = 1) \text{ or } (d_{k+2} = 1)),$$

In multi-level skeletonization, erosion is performed replacing the pixel with the minimum of its 4 nearest neighbours (b_0, b_2, b_4 or b_6).

In each iteration, we have to take care not to erode *col* pixels or pixels whose convexity is greater than the given threshold K_T.

Usually (Riazanoff et al., 1988), a col or saddle point is "a point P which displays in its neighbourhood at least two groups of contiguous points located lower than itself, as well as two groups located higher than itself". This definition may be refined using the crossing number: a col is a point whose crossing number (γ_4 in 4-connected neighbourhood or γ_8 in 8-connected neighbourhood) is greater than 1.

Example 1:

```
83 67 63          1 0 0
78 71 75   Δ→     1 1 1
68 70 90          0 0 1
4-connected             8-connected
  φ4 = 2                  φ8 = 4
  γ4 = 2                  γ8 = 2.
```

Example 2:

```
20 10  7          1 0 0
15 16 17   Δ→     0 1 1
16 15 18          1 0 1
4-connected             8-connected
  φ4 = 1                  φ8 = 4
  γ4 = 1                  γ8 = 3.
```

The point of example 2 is a col in 8-connected neighbourhood but not in 4-connected neighbourhood.

As in the previous section, convexity is roughly estimated by the maximum number of contiguous points lower than the central point.

Example:

```
25 17 20          1 1 1
14 13 15   Δ→     1 1 1
13 10  8          1 0 0
convexity:  K = 2.
```

Algorithm

Two images are processed by the algorithm. One corresponds to the result in the preceding iteration (Ima(i,j)), the other to the current image (Ima'(i,j)).

4-connected skeletonization

1. Preceding and current images are initialized to original image values
2. Repeat
   3. For each pixel (i,j) do
      4. If the pixel is not a local minimum ($\varphi_4 \geqslant 1$)
         5. If the convexity is *less than* the threshold ($K < K_T$)
            6. If the pixel is not a *col* in preceding image ($\gamma_4 \leqslant 1$)
               7. If the pixel is not a *col* in current image ($\gamma'_4 \leqslant 1$)
                  8. Then erode this pixel (Ima'$(i,j) \leftarrow$ Min (b_0, b_2, b_4, b_6))
               endif
            endif
         endif

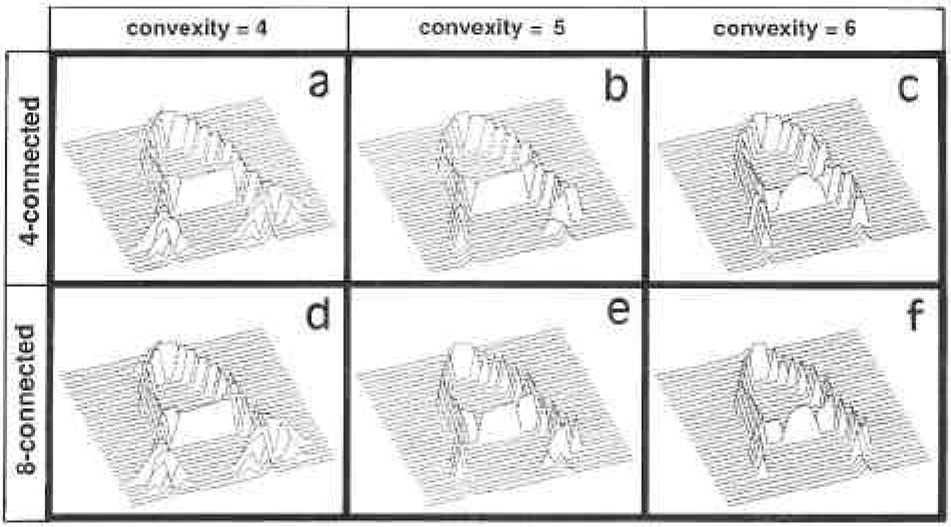

Figure 6. Multi-level skeleton (from Figure 5) achieved in 4-connected neighbourhood (a,b,c) and in 8-connected neighbourhood (d,e,f) and for several values of the threshold K_T = 4,5 and 6.

endif
enddo
9. Current image becomes preceding image (Ima ← Ima')
Until any one pixel might be modified in current iteration.

The 8-connected algorithm is achieved by replacing the γ_4 and γ'_4 tests of statements 6 and 7 by γ_8 and γ'_8 tests.

In comparison to the binary skeletonization, multi-level skeletonization is more time consuming because all the pixels have to be processed in each iteration. Given an image whose size is L*1, theoretically up to $(L+1)$ iterations may be necessary before invariance is reached. This corresponds to the step number required to propagate the local minimum effects along the greatest size of the image (i.e. the 4-connected diagonal length).

Results

Bench mark (Figure 5) results from 3 convolutions on the binary image 'A' processed in the previous section. The smoothing filter used is:

1 2 1
2 4 2
1 2 1.

Multi-level skeletonization was performed in 4-connected neighbourhood (Figures 6a, 6b and 6c) and in 8-connected neighbourhood (Figures 6d, 6e and 6f). In both, the K_T convexity threshold was set to values 4, 5 and 6.

Similarities between these results and those obtained through binary skeletonization (Figures 4b, 4c and 4d in 4-connected neighbourhood and 4g, 4h and 4i in 8-connected neighbourhood) must be emphasized. In spite of distortion introduced by the smoothing, a classical thresholding of those images produces skeletons very close to the latter ones shown in Figure 4.

N.B. Edge effects are due to an image edge initialization (set to maximal value) in order to keep the ridges possibly connected to the image edge.

5. Conclusions

The rationale about the curvature skeleton formalizes and extends other work of the authors. Improving the convexity estimator, it may be possible to determine some essential skeleton features from the first object contour examination.

Using some aspects of this new rationale, we conceived a parametrisable skeletonization procedure for binary and multi-level images.

This new method allows the user to control better the skeletonization process in order to obtain patterns best suited to his needs.

Multi-level skeletonization may be a very interesting tool. For instance, applied to the Digital Elevation Model, it gives a new solution to the ridge and valley lines extraction problem. By controlling the iteration number, this process allows us to enhance the sharpness of ridge lines and to erode massive 'patterns' while safeguarding their 'connectivity'.

References

Groen, F.C.A. and N.J. Foster (1984). A fast algorithm for cellular logic operations on sequential machines. *Pattern Recognition Letters* 2, 333–338.

Hilditch, C.J. (1969). Linear skeletons from square cupboards. In: B. Meltzer and D. Mitchie, Eds., *Machine Intelligence* 4. University Press of Edinburgh, 404–420.

Martinez-Perez, M.P., J. Jimenez and J.L. Navalon (1987). A thinning algorithm based on contours. *Computer Vision, Graphics and Image Processing* 39, 186–201.

Pavlidis, T. (1980). A thinning algorithm for discrete binary images. *Computer Graphics and Image Processing* 13, 142–157.

Shapiro, B., J. Pisa and J. Slansky (1981). Skeleton generation from *x, y* boundary sequences. *Computer Graphics and Image Processing* 15, 136–153.

Riazanoff, S., B. Cervelle and J. Chorowicz (1988). Ridge and valley line extraction from digital terrain models. *Int. J. Remote Sensing* 9, 1175–1183.

Van Vliet, L.J. and J.H. Verwer (1988). A contour processing method for fast binary neighbourhood operations. *Pattern Recognition Letters* 7, 27–36.