Finding grey-skeletons by iterated pixel removal

Carlo Arcelli and Giuliana Ramella

The grey-skeleton is understood as a connected subset of a grey-scale pattern, which is a stylized version consisting of a network of digital lines centrally placed along local higher intensity regions. We present a parallel thinning algorithm that relies on the iterated erosion of the pattern, and which proceeds from lower grey values towards higher ones until the grey-skeleton is finally obtained. The process includes a preliminary phase in which the significance of the hollows and plateaux possibly existing in the pattern is investigated. In particular, the hollows with a significant depth are regarded as topological constraints for the skeleton structure.

Keywords: grey-scale image, ridge, hollow, removal operation, parallel thinning, grey-skeleton

In structural pattern recognition, thinning algorithms are often a useful tool to represent a digital pattern by means of a stylized version, consisting of a set of digital lines that highlight the significant features of the original shape.

The design of thinning algorithms has mostly used binary images[1], on the assumption that these could adequately represent the real grey-scale images to be examined. However, recently there has been increasing interest in applying thinning directly to grey-scale images, motivated by the desire of processing inputs characterized by meaningful information distributed over different levels of intensity (as in the case of biomedical specimens). Furthermore, for inputs that can be reasonably understood as binary (e.g. in character recognition), a processing of the grey-scale digital image also seems more convenient in order to avoid shape distorsions that may irremediably affect the presence of features in the binary image generated, even if adaptive thresholding is taken into account[2,3].

Thinning of a grey-scale image may concern either the whole image or a subset of it; in this paper, we limit ourselves to discussing the latter. We consider grey-scale images where the regions of interest can be interpreted as constituting a multi-valued foreground emerging from a single-valued background, the latter being identified by suitably segmenting the digital image.

An analogy often done in the literature is to regard a grey-scale image as a terrain elevation map (the grey-value of a pixel being interpreted as its height). Accordingly, the identification of the set resulting after thinning (we call it a grey-skeleton, or a skeleton for short) has been related to the detection of topographical features such as ridges, peaks, saddles, and so on.

For instance[3,4], the grey-skeleton has recently been found by considering the image as a continuous surface in 3D Euclidean space, and by using the first and second partial derivatives of this surface to assign the proper topographical label to each pixel. In other cases[5-10], one has followed the traditional approach based on the repeated application of removal operations, eroding the grey-scale pattern until only a one-pixel-thick subset is obtained, mainly located in correspondence with its ridges. In this paper, we adopt the elevation map model and follow the latter approach, as this is more amenable to implementation on a parallel processor.

A major problem in using removal operations is how to find the ridge lines when the terrain elevation map is characterized by hollows, possibly close to each other (for instance, in case of an input pattern interpreted as adjacent craters – see *Figure 1*). In fact, since by definition hollows are surrounded by ridge lines, and removal operations are not expected to remove pixels belonging to ridge lines (that should be part of the resulting skeleton), it happens that pixels located inside hollows cannot be accessed, and some region of the pattern cannot be reduced to a one-pixel-thick subset. Although this problem has been discussed in a classical paper[11], and revisited some years later[12], until now a viable solution has not been found. In fact, most authors have focused on simple grey-scale images, with only one ridge line placed nearly centrally in the pattern, and with the grey-value gradually decreasing from the ridge towards the background. Indeed, even if a discussion on the topic is beyond the scope of this paper, we note that the literature does not yet provide

Istituto di Cibernetica, CNR, 80072 Arco Felice, Napoli, Italy
Paper received: 23 February 1994; revised paper received: 29 July 1994

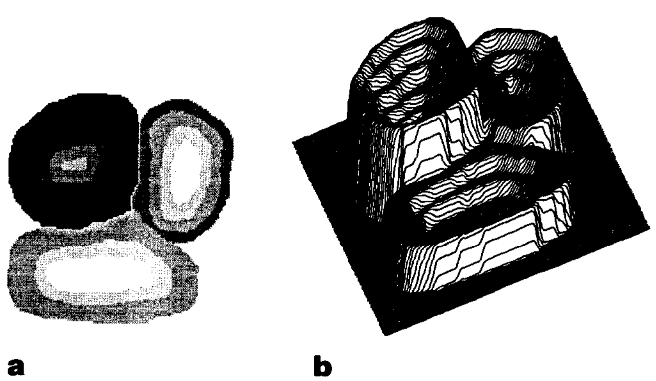


Figure 1 (a) Grey-scale pattern including adjacent hollows; (b) its topographical representation

any satisfactory insight into the matter of thinning non trivial grey-scale images, and there is no general agreement (contrary to the case of binary images) on what the requirements for a skeleton should be in order to constitute a meaningful representation.

The procedure we describe permits one to handle grey-scale patterns, including hollows, provided that their significant depth in the problem domain is known, and that it produces one-pixel-thick skeleton components that tend to be placed in correspondence with the midlines of the constant-grey-value elongated regions, as well as of the locally higher intensity regions.

The proposed parallel algorithm is based on the use of parallel operations which mostly need a 3×3 support, and are derived from those developed for binary images in the framework of cellular logic techniques[13]. We show experimental data obtained in correspondence with two different types of parallel removal operations. The first operation[14] requires a 3×3 support, and is an early explicit formulation of the well-known simple-point condition[15]. To produce correct thinning, it needs to be implemented in four subiterations, during each of which the arguments in the mask describing the operation are rotated by 90°. The second operation[16] requires the support of 11 pixels, and is one of the most sophisticated currently available. Though its use implies communication overheads in parallel computers, it does not need subiterations and it has been shown to be near optimally fast. In both implementations, the resulting grey-skeleton is a one-pixel-thick subset of the pattern.

The paper is organized as follows. The next section includes notations and basic notions; parallel removal operations are then introduced. The method is then outlined and the phases of the process described. Finally, the grey-skeletons obtained are shown in correspondence with test patterns characterized by different grey-scale distributions.

NOTATIONS AND BASIC NOTIONS

Let us consider a grey-scale digital image on the square grid, where the pixels are assigned one out of a finite number of values, increasing from value 0. The set of pixels with value 0 constitutes the background B, while every connected set of pixels with positive value constitutes a component of the foreground F.

The 8-connectedness and 4-connectedness are respectively chosen for F and for B. For the sake of simplicity, F is regarded as constituting only one component. Pixels are denoted by letters, which also indicate the corresponding grey-value.

The neighbours of a pixel p in F are its 8-adjacent pixels. They constitute the neighbourhood $N(p)$ of p, and are denoted by $n_1, n_2, \ldots, n_8$, where the subindexes increase clockwise from the pixel n_1 placed to the left of p:

$$\begin{matrix} n_2 & n_3 & n_4 \\ n_1 & p & n_5 \\ n_8 & n_7 & n_6 \end{matrix}$$

If $p > n_i$, for every i, p is termed a *peak point*.

A pixel of F with at least one $n_i = 0$, i odd, is termed a *border point*. An *end point* is a border point with only one neighbour n_i or two neighbours (n_i, n_{i+1}) in F.

When each n_i is understood as a Boolean variable equal to 1 if n_i belongs to F, and equal to 0 otherwise, and the variables are used in the expression to compute[17] the 8-connectivity number $C_8(p)$ as defined below, the resulting value $C(p)$ is equal to the number in $N(p)$ of 8-components of F:

$$C_8(p) = \Sigma_{i\,\mathrm{odd}}((1 - n_i) - (1 - n_i)\cdot(1 - n_{i+1})\cdot(1 - n_{i+2}))$$

When each n_i is understood as a Boolean variable equal to 1 if $n_i \geqslant p$, and equal to 0 otherwise, and the variables are used in the expression to compute the 8-connectivity number $C_8(p)$ previously defined, the resulting value $C^*(p)$ is equal to the number in $N(p)$ of 8-components of pixels with grey-level not less than p.

A pixel p, for which $C^*(p) > 1$, is termed a *ridge point*.

End points, peak points and ridge points are termed *topographical feature points*.

A path between two pixels p and q in F is a sequence of pixels $p_0, p_1, \ldots, p_{s-1}, p_s$, such that $p_0 = p$, $p_s = q$, and p_i is a neighbour of $p_{i-1}, i = 1, s$.

A flat region at elevation k is a connected set of pixels with a grey-value k.

A 4-connected flat region at elevation k, from which any path to pixels not in the region necessarily includes pixels with grey-value greater than k, is termed a *hollow* at elevation k. It is denoted by H_k.

The depth of H_k is the difference in grey-value $(h - k)$, where h is the minimal among the grey-values of a pixel p_r in any i-th path from H_k to pixels not in the region, such that $p_0 \leqslant p_1 \leqslant \ldots < p_s \leqslant \ldots \leqslant p_{r-1} \leqslant p_r > p_{r+1}$, for p_0 in H_k and p_s not in H_k.

An 8-connected flat region at elevation k from which any path to pixels not in the region necessarily includes pixels with grey-value smaller than k is termed a *plateau* at elevation k. It is denoted by PL_k. A peak point is a particular case of a plateau.

The height of PL_k is the difference in grey-value

$(k - h)$, where h is the maximal among the grey-values of a pixel p_r in any i-th path from PL_k to pixels not in the region, such that $p_0 \geqslant p_1 \geqslant \ldots > p_s \geqslant \ldots \geqslant p_{r-1} \geqslant p_r < p_{r+1}$, for p_0 in PL_k and p_s not in PL_k.

A ridge line is a one-pixel-thick connected subset of F mainly including pixels which are detected as topographical feature points.

The grey-skeleton of F is a connected set of ridge lines whose connectivity order is generally greater than that of F. Besides the loops in correspondence with the holes of F, the grey-skeleton has loops in correspondence with all the hollows of F whose depth is regarded as significant in the problem domain.

PIXEL REMOVAL

The scope is to obtain the grey-skeleton by repeatedly applying local removal operations. According to the skeleton model chosen, removal should regard neither topographical feature points nor pixels necessary to guarantee skeleton connectedness. The latter pixels will be retained by the removal operation we use (which is basically one used in thinning binary images). As a result, a pixel cannot be removed if this causes either a local disconnection of F or the creation of a hole in F. The first possibility is prevented by a number of neighbourhood conditions, while the second is trivially avoided by taking as candidates for removal only the border points.

In the case of binary images, all the border points are candidates for removal because the skeleton must be found along the central region of the pattern, as the result of a symmetric erosion. In the case of grey-scale images, the skeleton branches should be located along the ridges, wherever they exist. Alternatively, they should be located centrally within any plateau or any elongated flat region. Thus, the detection of the first type of branches can be achieved if one successively removes the sets of pixels with an increasing grey-value until only the sets with a locally higher grey-value (the ridges) are left. In turn, detection of the second type of branches requires that pixel removal be performed symmetrically, hence all the border points are candidates for removal.

For the above reasons, we accomplish the pixel removal phase as follows. The removal operation is applied to all the border points, and removal of part or all of them uncovers previously internal pixels, which in turn become border points. The grey-values of the removed pixels are ranked in ascending value, and are successively used on the current border to guide the removal of the newly born border points.

The current border points are interpreted as elements of sets of equal-grey-value pixels. The removal operation is applied on these sets, which are examined in increasing order, set by set, consistently with the ascending grey-values previously ranked. When all the sets of pixels corresponding to the ranked grey-values have been exhausted, the currently available border is taken into account in its entirety, and the previous process is iterated.

Parallel removal

A local operation is an operation that changes the grey-value of each pixel p in terms of its own grey-value and of that of a number of pixels close to p. The set of pixels whose grey-values constitute the arguments of the operation is called the *support of the operation*. A local removal operation is a local operation that is likely to change the grey-value of any p in F to 0. In the case of binary images, a local removal operation is adequate to perform a correct thinning if its application does not modify the number of components in both the foreground and the background of the image.

A parallel operation is an operation which is applied simultaneously to all the pixels of the image, and is one such that the new grey-value of each pixel does not depend upon the new grey-values of the remaining pixels.

To thin F, we take into account two implementations, based on either of the two removal operations R1 and R2, respectively derived from operations designed for thinning binary images in four subiterations[14] and in a fully parallel mode[16], and we refer to suitable binarizations of the grey-values of the pixels in the support.

Parallel operation R1

Let a pixel p in F be termed a north (east, south, west, respectively) border point if its neighbour $n_3 (n_5, n_7, n_1$, respectively) is 0. Moreover, let each n_i be regarded as a Boolean variable with $n_i = 1$ if n_i belongs to F, and $n_i = 0$ if n_i belongs either to the current background or to a significant hollow.

By operation R1 we denote the sequence of four parallel operations, each operation removing from F any p which satisfies the condition (α_k AND β), respectively for $k = 1$ to 4, and is not a topographical feature point, where $\alpha_1 \ldots \alpha_4$ and β are:

α_1 p is a north border point and $n_1 \cdot (1 - n_7) \cdot n_5 = 0$
α_2 p is an east border point and $n_3 \cdot (1 - n_1) \cdot n_7 = 0$
α_3 p is a south border point and $n_5 \cdot (1 - n_3) \cdot n_1 = 0$
α_4 p is a west border point and $n_7 \cdot (1 - n_5) \cdot n_3 = 0$
β $(1 - n_1) \cdot n_2 \cdot (1 - n_3) + (1 - n_3) \cdot n_4 \cdot (1 - n_5) + (1 - n_5) \cdot n_6 \cdot (1 - n_7) + (1 - n_7) \cdot n_8 \cdot (1 - n_1) = 0$

Thinning is accomplished in a quasi-isotropic fashion by repeatedly applying the sequence of the four parallel operations for p being a north, east, south, west border point, respectively. The process terminates when no pixel is removed during a whole sequence.

Parallel operation R2

Let p be a border point, and regard its neighbours as Boolean variables defined as stated with reference to operation R1. Moreover, let A(p) be the number of 01 transitions in N(p) encountered while going around p for a complete cycle[18].

R2 is the parallel operation that removes from F any p which is not a topographical feature point, and either:

(i) A(p) = 1, and the neighbourhood of p does not satisfy any of the following:

$$\begin{matrix} .0. \\ 1p1 \\ 111 \\ .0. \end{matrix} \quad \begin{matrix} .11. \\ 01p0 \\ .11. \\ \end{matrix} \quad \begin{matrix} .000 \\ 01p0 \\ .110 \\ ..0. \end{matrix} \quad \text{or}$$

(ii) the neighbourhood of p satisfies either of the following:

$$\begin{matrix} .00 \\ 1p0 \\ 01. \end{matrix} \quad \begin{matrix} 00. \\ 0p1 \\ .10 \end{matrix}$$

where the dot stands for don't care.

Thinning is accomplished in an isotropic fashion by repeatedly applying R2. The process terminates when no further pixel can be removed.

METHOD

We require that connectedness be preserved while repeatedly removing border points from the grey-scale pattern. Moreover, the skeleton should have branches that originate from end points and which are placed in correspondence with elongated regions and higher intensity regions.

Smoothing

In the binary case it is often convenient that some filtering precedes thinning, since the skeleton's structure is greatly affected by the 'salt and pepper' noise. The same holds in the grey-scale case. Meaningless skeletons are generally produced when F is characterized by topographical features such as narrow (at most two pixels wide) peaks and hollows, because these configurations have a size that is too negligible to regard their influence on the skeleton structure as significant. To remove these configurations, and to cause a general pattern smoothing, we use shrink and expand parallel operations based on the known peculiarities of the min and max functions[19]. Precisely, first the grey-value of every pixel p is replaced by the minimum grey-value of itself and its four n_i (i odd); then the grey-value of p is replaced by the maximum grey-value of itself and its eight neighbours. A consequence of this smoothing process is that topographical feature points (as defined above) are no longer present in the input image; they will appear only after pattern modifications caused by iteratively applying the removal operation.

Hollows

Ridge points cannot be removed because they are necessary to ensure perceptual significance of the skeleton branches; as a consequence, removal operations are not effective on pixels internal in the pattern and surrounded by a closed ridge line. This implies that pixels placed in hollows should be directly assigned to the background in order to prevent thick regions from remaining in the skeleton. On the other hand, this assignment should be done only for hollows having a significant depth. For any of the remaining ones, their level should be increased enough to force a change in the neighbourhood of some of the pixels, previously ridge points surrounding the hollow itself. In this way, those pixels will no longer be ridge points and they can be removed, so allowing removal of the pixels initially in the (now filled-in) hollow.

To detect hollows, the pixels p belonging to any flat region, and such that $p \leqslant n_i$, i odd, are first identified by a suitable label. Then, if the flat region is not a hollow, there will exist some pixel q with the same grey value as $p (q = p)$, but not labelled (i.e. $q > n_i$, for at least an n_i, i odd). The pixels as q show that the flat region is not a hollow. They are used as seeds from which to propagate through successive neighbours n_i, i odd, the signal 'this is not a hollow' over the labelled pixels p, from which the label is consequently removed. Propagation is done by iterative nearest neighbour signalling. Finally, all those pixels that have received the propagation signal have their label removed, so that every connected component of the labelled pixels still remaining in F is a hollow.

Let the integer h_1 be the significant depth of the hollows. This means that the hollows with a depth not smaller than h_1 should be regarded as components of the background, while the remaining hollows should be repeatedly filled-in until their depth decreases to zero. Their depth is decreased $h_1 - 1$ times by increasing $h_1 - 1$ times the grey-value of the pixels in each hollow. Hollow detection is repeated after each unit increase to check whether significant hollow still survive. In fact, the growth of $h_1 - 1$ units without any check would transform any shallow hollow with a depth $\ll h_1$ into a high plateau.

Plateaux

End points are the pixels from which skeleton branches are grown. To ensure some representativeness to the branches, each end point should be detected either in correspondence to a suitable curvature maximum of the outline of a constant-grey-level and elongated pattern subset, or in correspondence to a sufficiently high plateau. Unfortunately, to determine the significance of an end point by using only local checks is a hard task, even in the case of binary images. Generally, one adopts end point definitions that lead to the growth of a number of 'noisy' branches, which are successively pruned during a post-processing phase driven by some global criterion of significance. Even if not described in this paper, the same post-processing is foreseen for our thinning procedure, and this equally applies to the removal of 'noisy' branches grown from end points found on plateaux. In the latter case, however, we do not detect the end points created by the erosion of low plateaux. In fact, using a technique similar to the filling-in of insignificant hollows, a preliminary step of the procedure is devoted to diminish, for an *a priori* given amount, the height of the plateaux so that the low ones disappear.

To detect plateaux, the pixels p belonging to any flat region, which are not border points and are such that $p \geqslant n_i$, are first identified by a suitable label. Then, if the flat region is not a plateau, there will exist some pixel q with the same grey value as $p(q = p)$, but not labelled (i.e. $q < n_i$, for at least an n_i). The pixels like q are used as seeds from which to propagate through successive neighbours n_i the signal 'this is not a plateau' over the labelled pixels p, from which the label is consequently removed. Every connected component of labelled pixels remaining in F is a plateau. To detect plateaux that are significant, i.e. not smaller than an integer h_2, the height of each plateau is decreased $h_2 - 1$ times by decreasing $h_2 - 1$ times, unit by unit, the grey-value of its pixels.

To preserve the representative power of the skeletal pixels, the initial grey-values of the pixels of the surviving significant plateaux are restored by adding $h_2 - 1$ units to their current grey-value. The end points created by the repeated removal of the pixels belonging to the surviving plateaux can be regarded as perceptually significant tips of skeleton branches.

The algorithm

The smoothed version of F undergoes the plateau flattening and hollow filling processes. Then, the removal operation is iteratively applied to border points with a suitable grey-value.

In detail, the removal operation changes to 0 every border point that is not an end point, a peak point or a ridge point, and whose removal does not disconnect F. The grey-values pertaining to the removed pixels are ranked in increasing order $g_1, g_2, \ldots, g_k, \ldots, g_t$ (with g_{k+1} not necessarily equal to $g_k + 1$), while the hollows left in the pattern and with a grey-value not greater than g_t are set to 0. If $t > 1$ and $g_{t-1} < g_t - 1$, the grey-value $g_{t-1} = g_t - 1$ is also taken into account. Application of the removal operation is then restricted to those pixels whose grey-value is one of the ranked grey-values. The removal operation is repeatedly applied first to all the current border points equal to g_1, then to all the current border points not greater than g_2, and so on until also all the current border points g_t are possibly removed. As a result, all the components of pixels with a grey-value g_i, $i = 1$, t, are examined (including the pixels surrounding the hollows assigned to the background), and the only pixels not removed are end points, peak points, ridge points and suitable pixels linking the previous pixels so as to ensure skeleton connectedness.

Summarizing, the algorithm can be outlined as follows:

1. Smooth F by using operations based on min and max functions.
2. Fill in shallow hollows and flatten low plateaux. Identify the grey-values of the significant hollows.
3. Apply the removal operation, rank the grey-values of the removed pixels.
4. If no border point is removed, go to (8).
5. Remove the hollows whose pixels have a grey-value not greater than the maximal ranked grey-value.
6. Apply the restricted removal operation repeatedly so as to successively remove, up to the maximally ranked grey-value, the sets of pixels with a grey-value not greater than each of the ranked grey-values.
7. If at least a border point is removed, go to (3).
8. If there are hollows, remove all of them and go to (3).
9. Stop and take the un-removed pixels as skeleton pixels.

Beautifier

To improve the shape of the skeleton, it is often convenient to straighten any short zigzag present in the skeleton branches and to remove one-pixel-long branches. In our case, these aesthetic defects mainly originate from the criterion that privileges as elements of the skeleton those pixels which are strictly placed along regions with a higher grey-value. The aesthetics is improved by orderly applying the three parallel local operations below to the image, including the set resulting after pixel removal:

1. The configuration on the left below (and those rotated by 90°) where $p = 0$, $a > 0$, $b > 0$, $c > 0$, is changed into the configuration on the right (and the rotated ones) where $q = \max(a, b, c)$:

$$\begin{matrix} . & a & . \\ b & p & 0 \\ . & c & . \end{matrix} \qquad \begin{matrix} . & a & . \\ b & q & 0 \\ . & c & . \end{matrix}$$

2. If $p > 0$, $A(p) = 1$ and p has at least two $n_i > 0$, then p is set to zero.
3. If $p > 0$, $C(p) = 1$ and p has at least two $n_i > 0$, then p is set to zero.

RESULTS AND CONCLUDING REMARKS

The examples in *Figures 2–5* show some patterns taken from a set of grey-scale images digitized at 16 levels used to test the performance of the algorithm. Darker pixels are in correspondence with higher intensity regions. The skeleton is represented by a set of white pixels superimposed over the original input. The results obtained using operations R1 and R2 are shown. In both cases,

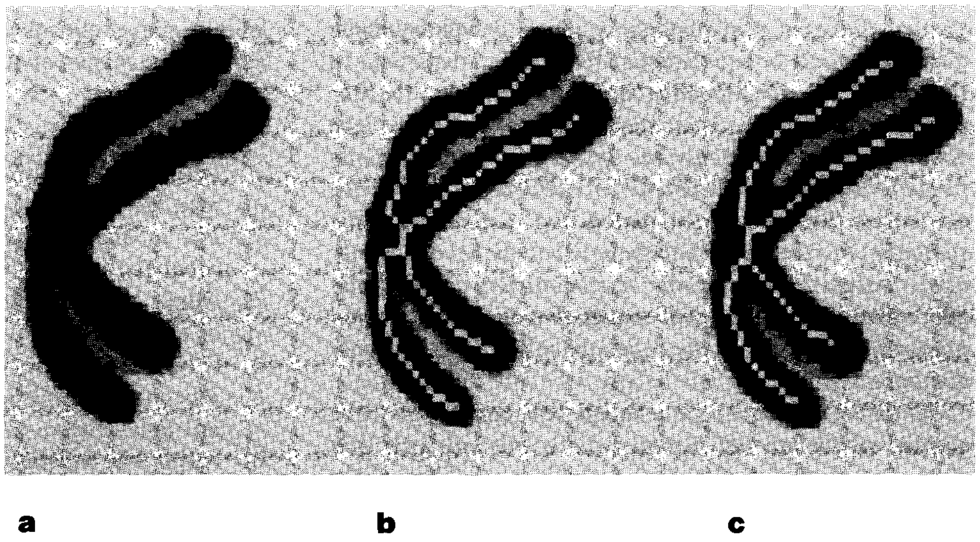


Figure 2 (a) Input pattern; (b) skeleton obtained by using operation R1; and (c) R2 superimposed over the input

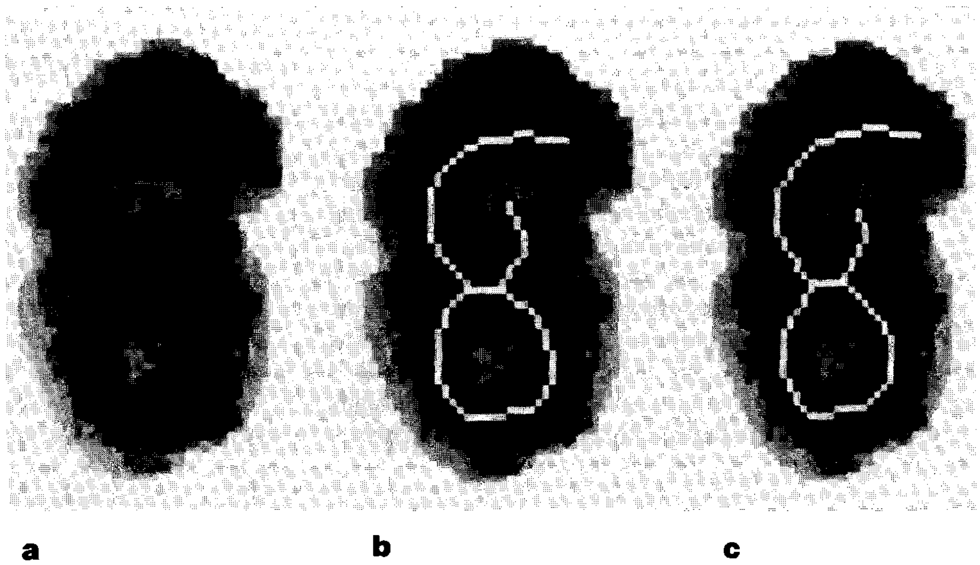


Figure 3 (a) Input pattern; (b) skeleton obtained by using operation R1; and (c) R2 superimposed over the input

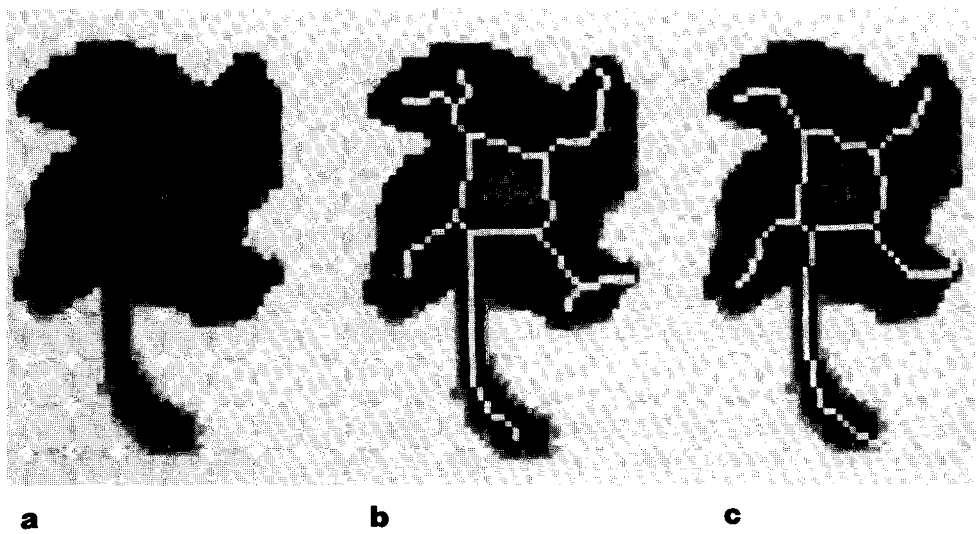
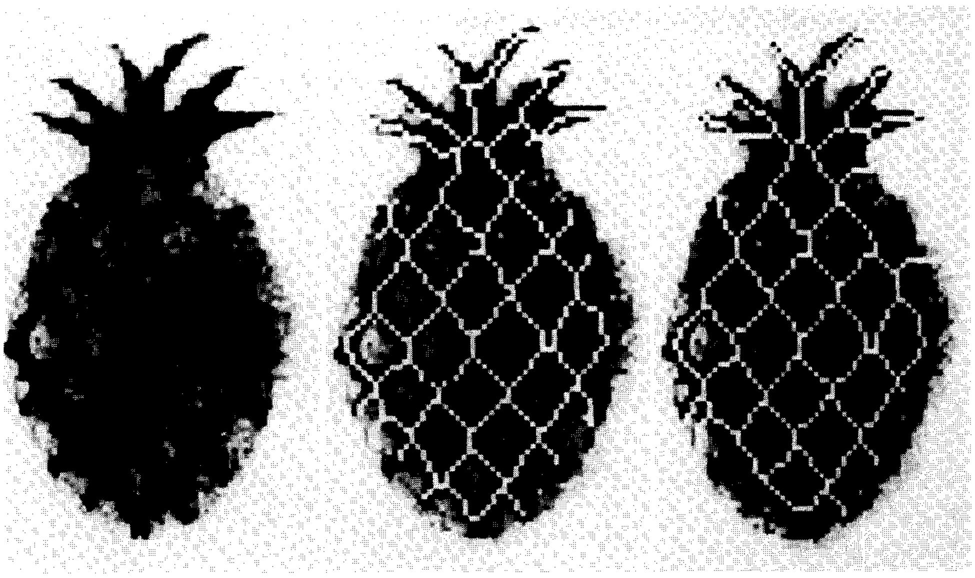


Figure 4 (a) Input pattern; (b) skeleton obtained by using operation R1; and (c) R2 superimposed over the input

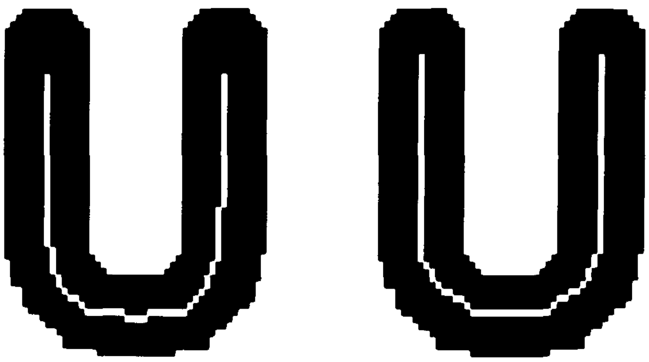
the geometry of the skeleton has been beautified. Note that when the pattern coincides with a plateau, i.e. the image is binary, the algorithm still produces meaningful results (see *Figure 6*).

As is well-known[16, 20], the reduced size of the support and the higher parallel speed are the properties that respectively better characterize R1 and R2. However, we note that when the procedure requires that the removal



a **b** **c**

Figure 5 (a) Input pattern; (b) skeleton obtained by using operation R1; and (c) R2 superimposed over the input



a **b**

Figure 6 Skeleton obtained by using operation R1 (a) and R2 (b) superimposed over the input

operation be used together with other operations (as in the case of thinning grey-scale images), using R1 rather than R2 may lead to an increase in computation time that is not so remarkable as when R1 and R2 are used almost alone (as in the case of thinning binary images). The aim of the comparison here is to show the quality of the results. Also referring to other experiments carried out, we observe that both operations produce largely similar skeletons (see *Figures 2* and *3*). A difference is that the algorithm based on R1 is more prone to the creation of skeleton branches than the algorithm based on R2 (*Figures 4* and *5*). This feature may constitute either an enrichment of the representative power of the skeleton or an undesirable redundancy. Which removal operation is preferable should be suggested by the class of images at hand, and by the available computing

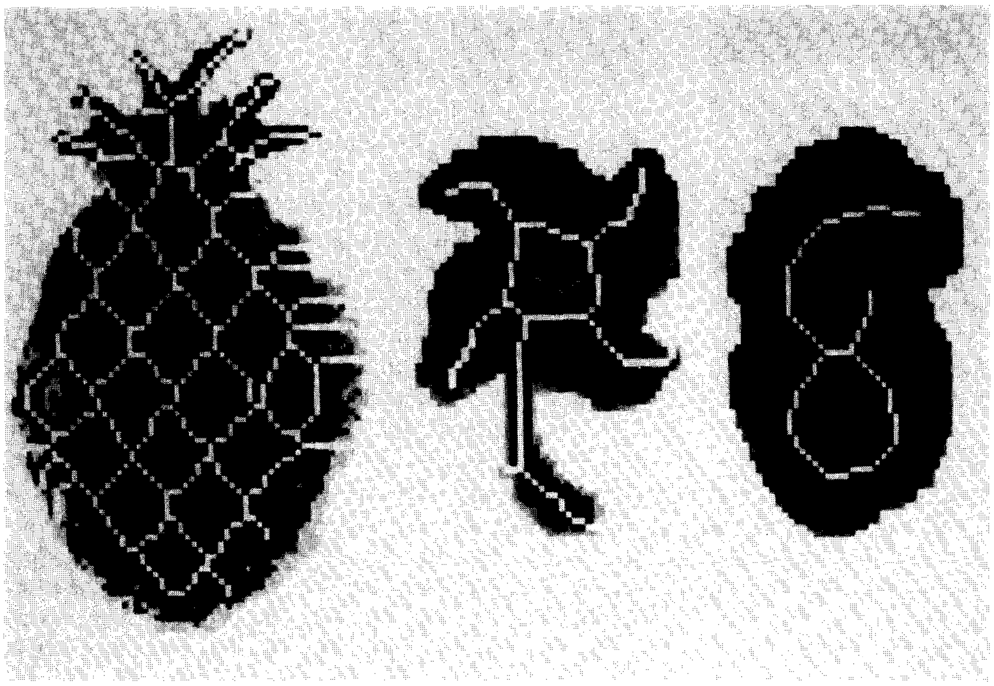


Figure 7 Skeletons obtained by using the removal operation described by Jang and Chin[21]

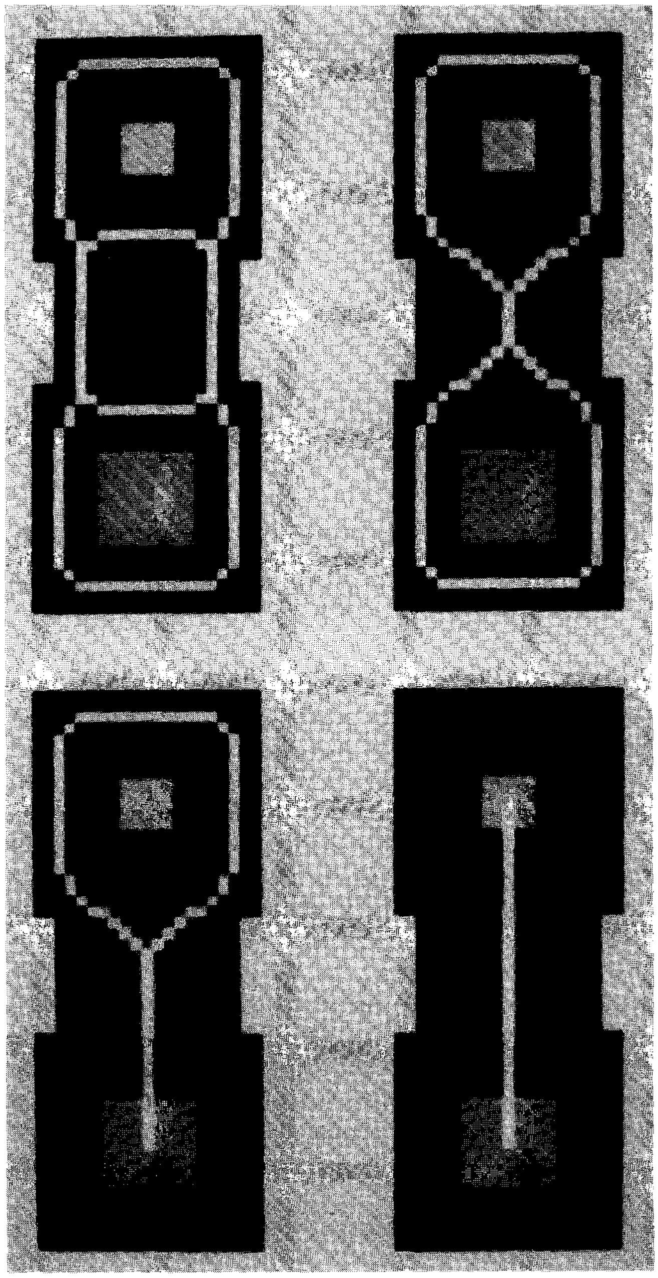


Figure 8 Number of skeleton loops diminishes as soon as the value of the significant depth of the hollows is sufficiently increased

machinery. For instance, in the case of implementation on a pipeline processing device, one might also refer to a recent parallel algorithm[21] with no subiterations, based on a removal operation consisting of the simultaneous application of a number of templates with sizes ranging from 3 × 3 to 5 × 5. The performance of that operation, when employed during the pixel removal phase of our algorithm, is shown in *Figure 7*.

The algorithm is articulated in four phases: smoothing, plateau and hollow processing, pixel removal and beautifying. One more phase, i.e. the pruning of insignificant peripheral skeleton branches, might have been considered to complete the thinning process, but it has not been discussed in the paper as significant pruning is largely problem dependent. Within each phase, one could have used alternative operations (equally tailored to the aim of the particular phase) as these are available in the vast repertoire of image processing techniques, and this could have produced a slightly different geometry of the resulting skeleton.

It seems of interest to remark that sometimes the requirement that the topographical feature points cannot be removed, coupled with the possibility for certain operations of removing pixels satisfying only given neighbourhood configurations, does not allow one to obtain a one-pixel-thick skeleton. For instance, consider the following pattern, constituted exclusively by pixels with a grey-value of c and d (with c < d), and use the removal operation R2:

```
d                                   d
   d  c  c  c  c  c  c  c  c  d
   d  d  d  d  d  d  d  d  d  d
d                                   d
```

This pattern cannot be thinned because the pixels labelled d are topographical feature points (either end points or ridge points) and the pixels labelled c do satisfy the first template in condition (i) of R2. As an additional shortcoming, note that in this case the parallel operations intended to beautify the skeleton are no longer adequate, since they cause a vanishing of two-pixel-thick horizontal and vertical configurations.

An important variation in the skeleton structure may only be introduced by the choice of the threshold value for the significant depth of the hollows. This peculiarity is illustrated in *Figure 8*, with reference to a pattern that includes hollows with a depth of 9, 6 and 7 (respectively in its upper, middle and lower parts). The number of loops diminishes as soon as the value of the significant depth is increased enough, until no loops in correspondence with hollows are left in the skeleton. In practical applications, this means that a sufficient knowledge of the problem domain allows one to give evidence to cavities understood as pathological features of a specimen, by representing them in terms of skeleton loops.

In our test patterns, the value of significance $h_l = 2$ was chosen for the hollows. This means that we discarded only those gentle hollows that are likely to be seen as insignificant by a human observer. In the absence of specific requirements from the task at hand, we regard this value of significance as appropriate for the range of scale we have dealt with. For the same reason, the significant height of the plateaux was also chosen equal to 2. Though requiring additional computational effort, an improvement worthy of interest regarding hollow detection could be to foresee more than one value for the significant depth, and to tune each value to a different elevation of the hollows.

In conclusion, the current algorithm originates a grey-skeleton located either centrally along constant-grey-value elongated parts of the pattern, or along ridges, possibly surrounding hollows with a significant depth. Pattern erosion is caused by successively applying a parallel removal operation to sets of pixels with higher and higher grey-values. Every set of pixels with a given grey-value is either ascribed to the background as a significant hollow (and a skeleton loop is created), or it is repeatedly shrunk by removing its current border points.

Contrary to the binary case, the skeleton obtained may possess a number of loops greater than the number of hole-components of the background. The choice of interpreting the hollows with a significant depth as topological constraints for the skeleton structure may be regarded as a step towards a more thorough understanding of the properties that should characterize a skeletal representation of a grey-scale pattern.

ACKNOWLEDGEMENTS

The help of Mr Salvatore Piantedosi in preparing the illustrations is gratefully acknowledged.

REFERENCES

1 Lam, L, Lee, S and Suen, C Y 'Thinning methodologies – A comprehensive survey', *IEEE Trans. PAMI*, Vol 14 (1992) pp 869–885
2 Preston, K and Duff, M J B *Modern Cellular Automata – Theory and Applications*, Plenum, New York (1984)
3 Wang, L and Pavlidis, T 'Direct gray-scale extraction of features for character recognition', *IEEE Trans. PAMI*, Vol 10 (1993) pp 1053–1067
4 Wang, L and Pavlidis, T 'Detection of curved and straight segments from gray scale topography', *CVGIP: Image Understanding*, Vol 58 (1993) pp 352–365
5 Dyer, C R and Rosenfeld, A 'Thinning algorithms for gray-scale pictures', *IEEE Trans. PAMI*, Vol 1 (1979) pp 88–90
6 Goetcherian, V 'From binary to grey tone image processing using fuzzy logic concepts', *Patt. Recogn.*, Vol 12 (1980) pp 7–15
7 Salari, E and Siy, P 'The ridge-seeking method for obtaining the skeleton of digital images', *IEEE Trans. Syst., Man & Cybern.*, Vol 14 (1984) pp 524–528
8 Yu, S S and Tsai, W H 'A new thinning algorithm for gray scale images by the relaxation technique', *Patt. Recogn.*, Vol 23 (1990) pp 1067–1076
9 Qian, K and Bhattacharya, P 'A template polynomial approach for image processing and visual recognition', *Patt. Recogn.*, Vol 25 (1992) pp 1505–1515
10 Abe, K, Mizutani, F and Wang, C 'Thinning of gray-scale images with combined sequential and parallel conditions for pixel removal', in *Visual Form Analysis and Recognition*, Arcelli, C, Cordella, L P and Sanniti di Baja, G (Eds.), Plenum, New York (1992) pp 1–10
11 Hilditch, C J 'Linear skeletons from square cupboards', in *Machine Intelligence IV*, Meltzer, B and Michie, D (Eds.), University Press, Edinburgh (1969) pp 403–420
12 Hilditch, C J 'Comparison of thinning algorithms on a parallel processor', *Image & Vision Comput.*, Vol 1 (1983) pp 115–132
13 Duff, M J B and Fountain, T J *Cellular Logic Image Processing*, Academic Press, New York (1986)
14 Arcelli, C 'A condition for digital point removal', *Signal Process.*, Vol 1 (1979) pp 283–285
15 Rosenfeld, A 'A characterization of parallel thinning algorithms', *Infor. & Control*, Vol 29 (1975) pp 286–291
16 Guo, Z and Hall, R W 'Fast fully parallel thinning algorithms', *CVGIP: Image Understanding*, Vol 55 (1992) pp 317–328
17 Yokoi, S, Toriwaki, J and Fukumura, T 'An analysis of topological properties of digitized binary pictures using local features', *Comput. Graphics & Image Process.*, Vol 4 (1975) pp 63–73
18 Rutovitz, D 'Pattern recognition', *J. Roy. Statist. Soc. A*, Vol 129 (1966) pp 504–530
19 Nakagawa, Y and Rosenfeld, A 'A note on the use of local min and max operations in digital picture processing', *IEEE Trans. Syst., Man & Cybern.*, Vol 8 (1978) pp 632–635
20 Hall, R W 'Fast parallel thinning algorithms: parallel speed and connectivity preservation', *Commun. ACM*, Vol 32 (1989) pp 124–131
21 Jang, B K and Chin, R T 'One-pass parallel thinning: analysis, properties, and quantitative evaluation', *IEEE Trans. PAMI*, Vol 14 (1992) pp 1129–1140