Pattern Recognition Letters 12 (1991) 235-240
North-Holland

April 1991

A parallel algorithm for decomposition of binary objects through skeletonization

S.K. Parui

Electronics and Communication Sciences Unit, Indian Statistical Institute, Calcutta-700035, India

A. Datta

Computer and Statistical Service Centre, Indian Statistical Institute, Calcutta-700035, India

Received 25 September 1990
Revised 9 January 1991

Abstract

Parui, S.K. and A. Datta, A parallel algorithm for decomposition of binary objects through skeletonization, Pattern Recognition Letters 12 (1991) 235-240.

A simple decomposition technique is proposed for binary objects. First a certain skeleton of a binary object is found and then the skeleton is decomposed into connected components. Finally a simple object part is reconstructed from each such skeletal component. The algorithms proposed for skeletonization and reconstruction are parallelizable.

Keywords. Binary objects, decomposition, skeletonization, connected components, parallel algorithms.

1. Introduction

Decomposition of a pattern or object into simpler and smaller subpatterns or parts is often useful in image processing and pattern recognition (Pavlidis, 1977). In this paper, we present a simple technique for decomposing a binary object into object parts which are easier to deal with. Here we consider objects which mainly consist of elongated parts. Character patterns are one such example. The decomposition here is based on a certain skeletonization of the object (Parui, 1984). The skeleton is first decomposed into several connected components from which the object parts are then reconstructed. The skeleton finding and reconstruction algorithms are implementable on parallel machines. The skeleton in the analog case is defined in the following way. For every point P in the object, all possible chords passing through it are considered and the chord with the minimum length is chosen. If P is the mid-point of the minimal chord, then P is a skeletal point. The skeleton of the object consists of all such skeletal points and is not necessarily connected for a connected object (Figure 1).

2. Skeletonization of a binary object

The input image here is represented as a 2-D binary matrix where the 0- and 1-pixels indicate the background and the object, respectively. For

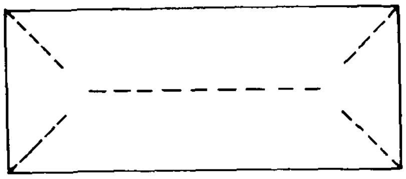

Figure 1. Dashed lines indicate the skeleton of a rectangular object in the analog case.

skeletonization we consider only the vertical and horizontal runs of 1-pixels. The *length* of such a run is $(n-1)$ where n is the number of pixels in the run (Figure 2). For a run of n 1-pixels, the *mid-pixel* of the run is defined as the $[(n+1)/2]$-th pixel from the start of the run (that is, the top pixel for a vertical run and the leftmost pixel for a horizontal run) (Figure 2).

Now the skeleton of the object is found in the following way. For every object pixel P we find the runs of 1's containing P only in vertical and horizontal directions (in the analog case all possible directions are considered). From among these two runs, we choose the one with the shorter run length. It is called the *minimal run* containing P. If P is the mid-pixel of this minimal run, then P is called a *skeletal pixel.* All such skeletal pixels constitute the skeleton. If for a skeletal pixel P, the length of the corresponding minimal run is m, then we say that the object has *thickness* m at P.

Thus the minimal runs are either vertical or horizontal. If the vertical and horizontal run lengths are equal, then the vertical run is considered first. If the corresponding pixel is not its mid-pixel, then the horizontal run is considered. Each skeletal pixel is given a label depending on the minimal run containing the pixel. A skeletal pixel emerging from a vertical minimal run has label V. In the other case, the label is H. Thus, the pixels with labels V and H constitute the entire skeleton (Figure 3b).

```
1 1 1 1 1 1      1 1 1 1 1      1
                                1
                                1
                                1
   (a)              (b)        (c)
```

Figure 2. The lengths of the runs in (a), (b) and (c) are 5, 4 and 3 respectively. The mid-pixels are underlined.

```
1 1 1 1 1 1 1 1 1 1 1 1 1        1 1 1 1 1 1 1 1 1 1 1 1 V
1 1 1 1 1 1 1 1 1 1 1 1 1        V V V 1 1 1 V V V V V V 1
1 1 1 1 1 1 1 1 1 1 1 1          1 1 1 1 1 1 1 1 1 1 1 1
      1 1 1                            1 H 1
      1 1 1                            1 H 1
      1 1 1                            1 H 1
    1 1 1 1                          1 H 1 1
    1 1 1 1                          1 H 1 1
  1 1 1 1                          1 H 1 1
  1 1 1 1                          V H 1 1
  1 1 1 1                          1 H 1 1
  1 1 1 1                          1 H 1 1

     (a)                                (b)

. . . . . . . . . . . . 2
3 3 3 . . . 3 3 3 3 3 3 .
. . . . . . . . . . . .
      . 3 .
      . 3 .
      . 3 .
    . 4 . .
    . 4 . .
  . 4 . .
  4 4 . .
  . 4 . .
  . 4 . .

     (c)
```

Figure 3. The original pattern is in (a). Its skeleton is shown in (b) where the pixels with labels V and H form the skeleton. The thickness map for the skeleton in (b) is shown in (c) where the non-skeletal pixels are shown as dots.

It can be seen that skeletal parts formed by the V-pixels indicate the horizontal parts (that is, parts making an angle between -45 and 45 degrees with the x-axis) of the object and those formed by the H-pixels indicate its vertical parts (that is, parts making an angle between 45 and 135 degrees with the x-axis). Apart from giving a label to every skeletal pixel, the thickness also is assigned to every skeletal pixel and is stored in what is called the *thickness map* of the skeleton (Figure 3c).

It is clear that a skeletal pixel can be found independently of other skeletal pixels and hence the skeleton finding can be parallelized. On the other hand, the skeleton thus found is not necessarily connected for a connected object. All the 8-connected components of the V-pixels and H-pixels present in the skeleton are found. These components will be used in the next section for decomposing the object into smaller parts.

A *digital curve* S is defined as an 8-connected set of pixels such that every pixel except two of S has exactly two 8-neighbours. The two pixels each having just one 8-neighbour are the end pixels of S.

Proposition 1. *Every 8-connected component in the skeleton as defined above is a digital curve* *provided the thickness at skeletal pixels at a crossing or branching is more than* 1.

(For a 1-pixel thick 'Y'-like pattern, all the object pixels form one single skeletal component which is not a digital curve. This type of patterns is not considered in this paper.)

3. Decomposition

Note that the skeleton does not remain connected at a crossing or a sharp turn or a joint. At these junctures among others, the object is disconnected for decomposition. First the skeleton is decomposed into connected components and then the object parts are reconstructed from these skeletal components. Details of the decomposition technique are given below.

After constructing the skeleton, we locate the 8-connected components of V's and H's. Each such skeletal component (which is a digital curve) corresponds to a single simple part of the original object, and these simple parts can be reconstructed from the skeletal components and the thickness map. This is possible by recovering the minimal run containing every skeletal pixel and taking the union of all such minimal runs. Note that this can be done in parallel since the reconstruction from one skeletal pixel does not affect the reconstruction from any other skeletal pixel.

It can be seen that the skeletal pixels do not span the whole object. In other words, there may be some object pixels or parts which do not fall on the minimal run of any skeletal pixel. In Figure 3, the skeleton is decomposed into four components from which the object parts can be reconstructed. But here the middle part of the top of the object is not spanned by the skeleton. (Though the skeleton contains information about the overall shape of the object.) Another problem is that there may be some object pixels which are spanned by more than one skeletal component. For example, in Figure 3b, the object pixels spanned by two V-pixels in the bottom left part are also spanned by the H-pixels. Both these problems during reconstruction can be overcome if a multi-pass (rather than one-pass as discussed above) reconstruction algorithm is used which runs as follows.

```
    1 H 1
    1 H 1
    1 H 1
  1 H 1 1
  1 H 1 1
1 H 1 1
1 H 1 1
1 H 1 1
1 H 1 1

   (a)
```

```
1 1 1 1 1 1 1 1 1 1 1 1 V
V V V V V V V V V V V V 1
1 1 1 1 1 1 1 1 1 1 1 1

          (b)
```

Figure 4. Above are the two parts after decomposing the object in Figure 3a using the multi-pass algorithm (Algorithm 1).

Algorithm 1

Step 1. Set A = the whole object.

Step 2. Find the skeleton of A in the way described in Section 2. Also, find the thickness map of the skeleton.

Step 3. Find the connected components of V- and H-pixels in the skeleton. Choose the component C that has the maximum length. If this length is not large enough, stop. Otherwise, go to Step 4.

Step 4. From C and the thickness map, reconstruct only that part of the object that corresponds to C. Call the reconstructed object part R.

Step 5. Remove the pixels of R from A. Set $A = A - R$. Go to Step 2.

Note that only Step 3 in the above algorithm is sequential while the other steps are parallel since they involve only local operations. As is clear from above, the number of passes in the multi-pass algorithm is the same as the number of the object parts that the algorithm produces. For example, for the object in Figure 3a, two passes are needed. In the first pass, the skeletal component of H-pixels is the component with the maximum length (Figure 3b). The reconstructed part R is shown in Figure 4a. In the second pass, there is only one skeletal component (Figure 4b) which indicates the upper horizontal part of the object. The one-pass algorithm on the other hand would have unfortunately produced more than two parts after decomposing the object in Figure 3a.

Now it should be noted that decomposing the skeleton into connected components using only the directional labels (V or H) may not be enough for

```
1 1 1 1 1 1 1 1 1               1 1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1 1               1 1 1 1 1 1 1 1 1
V V V V V V V V V V V V V V V V V V V V V V V V V
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1 1               1 1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1 1               1 1 1 1 1 1 1 1 1
```

(a)

```
. . . . . . . . .               . . . . . . . . .
. . . . . . . . .               . . . . . . . . .
6 6 6 6 6 6 6 6 6 2 2 2 2 2 2 2 6 6 6 6 6 6 6 6 6
. . . . . . . . . . . . . . . . . . . . . . . . .
. . . . . . . . .               . . . . . . . . .
. . . . . . . . .               . . . . . . . . .
```

(b)

```
1 1 1 1 1 1 1 1 1                   1 1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1 1                   1 1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1 1    1 1 1 1 1 1 1  1 1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1 1    1 1 1 1 1 1 1  1 1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1 1                   1 1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1 1                   1 1 1 1 1 1 1 1 1
```

(c)

Figure 5. (a) An object where its skeleton (V-pixels) has only one connected component. (b) The thickness map of the skeleton in (a) where there are two pairs of adjacent pixels with thickness difference 4. The skeletal component is decomposed into subcomponents at these two pairs of pixels. (c) The object is correspondingly split into three object parts.

```
1 1 1 1 1 1 1 1 1 1           1 1 1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1 1 1 1 1   1 1 1 1 1 1 1 1 1 1 1 1
V V V V V V V V V V V V V V V V V V V V V V V V V
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1 1 1 1       1 1 1 1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1 1               1 1 1 1 1 1 1 1 1
```

(a)

```
. . . . . . . . .               . . . . . . . . .
. . . . . . . . .               . . . . . . . . .
6 6 6 6 6 6 6 6 6 5 4 3 2 3 4 5 6 6 6 6 6 6 6 6 6
. . . . . . . . . . . . . . . . . . . . . . . . .
. . . . . . . . .               . . . . . . . . .
. . . . . . . . .               . . . . . . . . .
```

(b)

```
1 1 1 1 1 1 1 1 1 1                  1 1 1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1 1 1 1 1          1 1 1 1 1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1 1 1 1 1 1        1 1 1 1 1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1 1 1 1 1 1        1 1 1 1 1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1 1 1 1              1 1 1 1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1 1                      1 1 1 1 1 1 1 1 1
```

(c)

Figure 6. (a) An object where its skeleton (V-pixels) has only one connected component. (b) The thickness map of the skeleton in (a) with a local minimum at thickness = 2 where the skeletal component is decomposed into two subcomponents. (c) The object is correspondingly split into two parts.

meaningful decomposition of the object (Figure 5). The thickness map can be useful in such cases. Since a skeletal component is a digital curve (from Proposition 1), its pixels can be scanned in a linear fashion. Through such a scan of the pixels their thickness values can be examined sequentially. If within a single skeletal component, two adjacent pixels have significantly different thickness, then the component is split into two between these two pixels (Figure 5b). But sometimes the difference in thickness is not significant around a skeletal pixel where the object should be split (Figure 6). In such cases the local minima in thickness values can be found. At these local minima the skeletal component can be split (Figure 6b).

Now Algorithm 1 can be modified to incorporate the thickness-based skeleton splitting criteria discussed above. The modified algorithm is given below.

Algorithm 2

Step 1. Set A = the whole object.

Step 2. Find the skeleton of A in the way described in Section 2. Also, find the thickness map of the skeleton.

Step 3. Find the connected components of V- and H-pixels in the skeleton. Split each such component into subcomponents using the thickness map as discussed above. Find the subcomponent C that has the maximum length. If this length is not large enough, stop. Otherwise, go to Step 4.

Step 4. From C and the thickness map, reconstruct only that part of the object that corresponds to C. Call the reconstructed object part R.

```
1 H 1
1 H 1
1 H 1
1 H 1
1 H 1
1 H 1
1 H 1
1 H 1
1 H 1
1 H 1
1 H 1
```

(a)

```
                H
              1 H 1
            1 1 V V V
          1 1 V 1 1
        1 1 V 1 1
      1 1 V 1 1
    1 1 V 1 1
  1 1 V 1 1
V V V 1 1
  1 H 1
    H
```

(b)

Figure 7. Two digital rectangles (vertically oriented in (a) and oriented at 45 degrees in (b)) have different skeletons.

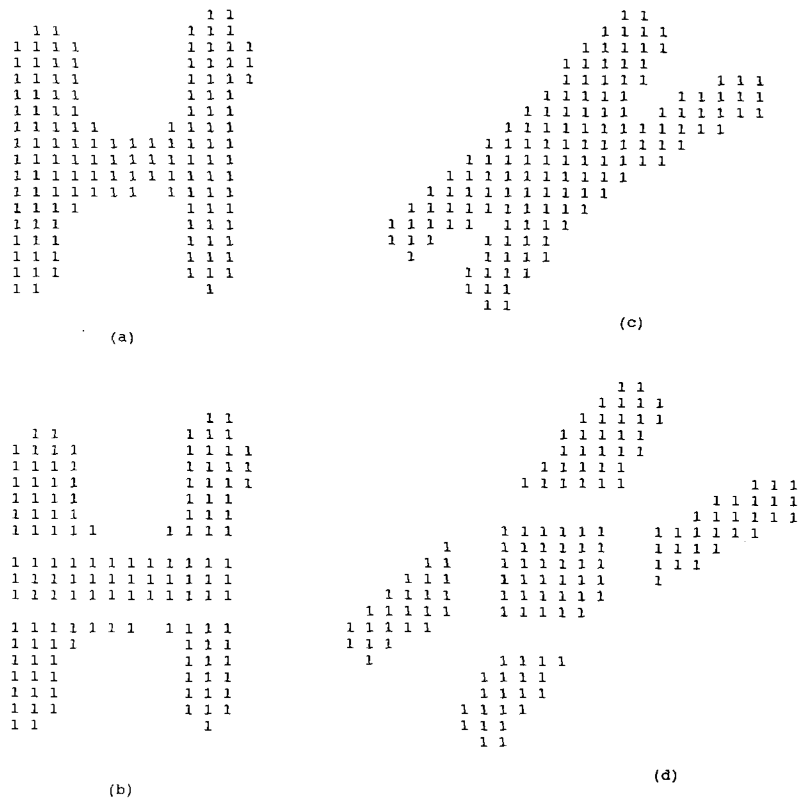

Figure 8. (a) A digital image of the character 'H' and (c) a digital image of a chromosome (taken from Rosenfeld and Kak (1982)). The decomposed parts of (a) and (c) using the multi-pass algorithm are shown in (b) and (d), respectively.

Step 5. Remove the pixels of R from A. Set $A = A - R$. Go to Step 2.

4. Conclusions

It can be seen that the skeletonization and reconstruction parts of the decomposition algorithm proposed in the present paper can be implemented on a parallel machine while finding the skeletal components and their lengths involves sequential computations.

The skeleton proposed above has the disadvantage of being dependent on the orientation of the input pattern (Figure 7). In the definition of a skeletal pixel in Section 2, only two directions (vertical and horizontal) are considered. If four directions (including the diagonal directions) are considered instead, the skeleton in many cases becomes 2-pixel thick and Proposition 1 no more holds.

The above skeletonization and decomposition algorithms can easily be extended to 3 dimensions. Instead of two directions (vertical and horizontal), three orthogonal directions are to be considered in 3 dimensions. There are two possible definitions of a skeletal pixel in this case. For every object pixel P, consider the three runs of 1-pixels (containing P) in these three directions. If P is in the middle of the minimal run, P is a skeletal pixel. The other possible definition is as follows. For every object pixel P, consider the three cross-sections consisting of 1-pixels (containing P) in the three directions. Choose the cross-section with the minimum area (that is, the minimum number of 1-pixels). If P is in the middle of the minimal cross-section, P is a skeletal pixel. In the case of the first definition, the skeleton will be a 1-pixel thick digital surface and in the case of the second, it will be a 1-pixel thick digital line in 3 dimensions.

The skeletal decomposition technique proposed above can have applications in character recognition, chromosome analysis, curve segmentation, etc. (Figure 8).

References

Pavlidis, T. (1977). *Structural Pattern Recognition.* Springer, Berlin.

Parui, S.K. (1984). Some studies in analysis and recognition of 2-dimensional shapes. Ph.D. Thesis, Indian Statistical Institute.

Rosenfeld, A. and A.C. Kak (1982). *Digital Picture Processing, Vol. 2.* Academic Press, New York.