Some geometric operations on binary pictures and their shape preserving properties

Swapan K. PARUI
Indian Statistical Institute, Calcutta 700035, India
Currently a Visiting Scientist in Institut für Strahlenschutz, G.S.F., Neuherberg, FRG

Abstract: Shape preserving algorithms for magnification by arbitrary factors and rotation by arbitrary angles of binary pictures in two dimensions are presented. These algorithms may be useful for shape matching. How the proposed algorithms preserve the shape of the input picture is described.

Key words: Binary pictures, magnification, rotation, topological properties, shape preservation.

1. Introduction

The geometric operations that are dealt with here are magnification and rotation of binary pictures in two dimensions. Two algorithms are proposed for these operations which preserve the shape in general and the topological properties in particular of the input picture and can be useful for shape matching (Parui and Dutta Majumder, 1984; Parui, 1984). (Shape in this paper means what remains of an object after disregarding its position, size and orientation.) For the magnification algorithm, the factor of magnification can be an arbitrary real number and is the same along the x- and y-axes. The angle of rotation in the rotation algorithm also can be arbitrary. The shape preserving properties of both the algorithms are described. The basic ideas behind the present magnification algorithm were given by Rosenfeld and Johnston (1970), though its shape preserving properties were not dealt with in detail.

A binary picture is represented as a binary matrix each of whose elements is either 0 or 1, 0's indicating the background and 1's the object. 4-connectedness is assumed for the background pixels and 8-connectedness for the object pixels. Each maximal 8-connected subset of 1-pixels is called a *component* of the picture. Each maximal 4-connected subset of 0-pixels surrounded by 1-pixels is a *hole* and the rest of the 0-pixels form the *background*. A binary picture consists of the background and at least one component which may or may not have holes in it. The first and last columns and the first and last rows of a binary picture consist only of 0-pixels. The *order of connectivity of a component A* is the number of (4-connected) components in the complement of A (Rosenfeld, 1970). It is in fact the number of holes in $A+1$. The *order of connectivity of a binary picture* is the sum of the orders of connectivity of all its components. In other words, it is the number of components + the number of holes present in the picture.

The *thinness of a component* is the minimum number of 1-pixels whose removal from it changes (reduces or increases) the order of connectivity of the picture. The *thinness of a hole* is the minimum number of 1-pixels whose incorporation into it changes the number of holes in the picture. (Note that a hole with exactly 2 0-pixels is of thinness 2.

Thinness of only such a hole contradicts our intuition, since this thinness should have been 1 intuitively.) The *thinness of the background* is the minimum number of 1-pixels whose incorporation into it changes the order of connectivity of the picture (Figure 1).

The algorithms for magnification and rotation of binary pictures along with their shape preserving properties are described in Sections 2 and 3 respectively. The proofs of these properties are not given but are available elsewhere (Parui, 1984). Conclusions are given in Section 4.

2. Magnification

When the shapes of two binary pictures with different areas are to be matched by some kind of template matching, one approach is to dilate the smaller picture so that it attains the area of the larger one without disturbing its shape. Such a shape matching can be done on the basis of area of mismatch (Parui and Dutta Majumder, 1984). For the magnification algorithm described below, the ratio of the above two areas can be any real number, not necessarily a perfect square or even an integer.

Algorithm 1. To magnify a binary picture by an arbitrary real factor $\text{FACT} > 1$.

From Figure 2 it is clear that each row of the magnified or output picture matrix, say N, partially contains one or two rows of the input matrix, say M. The same is true for the columns also. Let the Ith row of N contain the $\text{IROW}(I)$th and

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

Figure 1. In the binary picture above, (i) the order of connectivity is 3 (2 components and 1 hole), (ii) the thinnesses of the connected components are (from left to right) 2 and 1, (iii) the thinness of the hole is 1 and (iv) the thinness of the background is 2. (In the figure above and in the subsequent figures, 0-pixels are suppressed.)

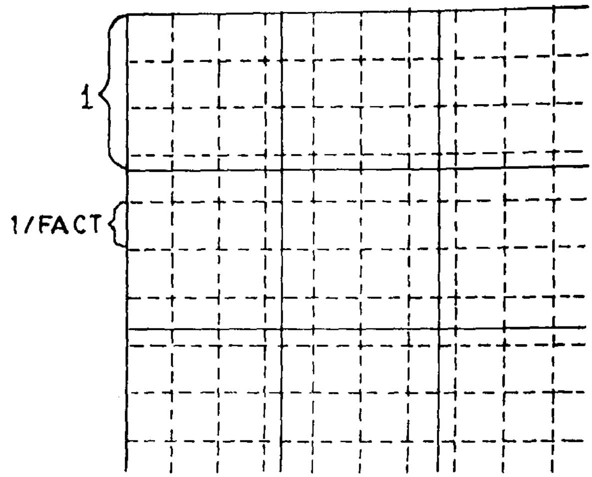

Figure 2. Pixels with unbroken lines and pixels with broken lines indicate the input and output pixels respectively. Here, the factor of magnification (FACT) is 10/3 and the sizes of the input and output pixels are 1 and 3/10 respectively.

$\text{IROW}(I)+1$st rows of M with weights $R_1(I)$ and $S_1(I)$ respectively, where $R_1(I)+S_1(I)$ is the (side) size of the pixels of the output picture assuming the size of the pixels of the input picture to be 1. Let the factor of magnification here be $\text{FACT} > 1$. That is, the size of the output pixels is $1/\text{FACT}$ which is less than 1. Suppose, for the Jth column of N, the corresponding variables are $\text{JCOL}(J)$, $R_2(J)$ and $S_2(J)$ where again $R_2(J)+S_2(J) = 1/\text{FACT}$.

Step 1 (Initialization). Set $P_1 = 0$, $P_2 = 1/\text{FACT}$, $I = 1$, $\text{IROW}(1) = 1$, $R_1(1) = 1/\text{FACT}$ and $S_1(1) = 0$.

Step 2. Set $I = I+1$, $P_1 = P_2$ and $P_2 = P_2 + 1/\text{FACT}$.
If there is an integer INT such that $P_1 < \text{INT} < P_2$, then set $\text{IROW}(I) = \text{INT}$, $R_1(I) = \text{INT} - P_1$ and $S_1(I) = P_2 - \text{INT}$.
Repeat Step 2.
If there is an integer INT such that $\text{INT} - 1 \leqslant P_1$, $P_2 \leqslant \text{INT}$, then set $\text{IROW}(I) = \text{INT}$, $R_1(I) = 1/\text{FACT}$ and $S_1(I) = 0$.
Repeat Step 2.
(Note that since $P_2 - P_1 = 1/\text{FACT} < 1$, either P_1 and P_2 lie between two consecutive integers or exactly one integer lies between P_1 and P_2.)
Stop when all the rows of the output matrix N are exhausted.

In the same way as above, $\text{JCOL}(J)$, $R_2(J)$ and $S_2(J)$ are computed for the output columns. The construction of the output matrix N is as follows:

For each row I and each column J of the output matrix N, let $X(I, J)$ equal

$$\begin{aligned}&R_1(I) * R_2(J) * M(\text{IROW}(I), \text{JCOL}(J))\\ &\quad + R_1(I) * S_2(J) * M(\text{IROW}(I), \text{JCOL}(J)+1)\\ &\quad + S_1(I) * R_2(J) * M(\text{IROW}(I)+1, \text{JCOL}(J))\\ &\quad + S_1(I) * S_2(J) * M(\text{IROW}(I)+1, \text{JCOL}(J)+1)\end{aligned}$$

where M is the binary input matrix. $X(I, J)$ indicates the area of the part of the output (I, J)th pixel that falls under 1-pixels of the input binary picture represented by the binary matrix M.

Make $N(I, J)=1$ if $X(I, J) \geqslant 1/(2 * \text{FACT} * \text{FACT})=$ half of the area of an output pixel, and make $N(I, J)=0$ otherwise.

Note that the above algorithm is flexible in the sense that the factors of magnification along the vertical and horizontal axes need not be the same. In other words, we can use $\text{FACT}_1 > 1$ and $\text{FACT}_2 > 1$ for the vertical and horizontal axes respectively, where FACT_1 and FACT_2 are not necessarily the same. In that case, $N(I, J)$ should be redefined as

$$\begin{aligned}N(I, J) &= 1 \quad \text{if } X(I, J) \geqslant 1/(2 * \text{FACT}_1 * \text{FACT}_2),\\ &= 0 \quad \text{otherwise.}\end{aligned}$$

But, in this paper, we are concerned with magnification with only uniform change of scale along the two axes. That is, FACT_1 and FACT_2 are taken to be the same because shape preservation is one of the main aims here. It is clear that magnification by Algorithm 1 does not always entirely preserve the shape of the input picture because of its digital nature. Distortions may be present near the boundary of the picture. But when the perimeter is small compared to the area of the picture, shape distortions will be low. An example of a magnified picture is given in Figure 3.

The shape preserving properties of magnification by Algorithm 1 are given below:

Property 1.1. *If a hole of a binary picture has only two 0-pixels, then after magnification with arbitrary factors, the hole does not disappear, that is, gives rise to at least one output 0-pixel and the resulting output 0-pixels (if more than one in number) are 4-connected.*

But if a hole of a binary picture has only one 0-pixel, then after magnification the hole may disappear provided the factor of magnification is close to 1. One such example is given in Figure 4.

```
    1 1
    1 1
    1 1 1
  1 1 1 1 1
  1 1 1 1 1 1 1
  1 1 1 1 1 1 1 1 1 1
  1 1 1 1 1 1 1 1 1 1
    1 1 1 1 1 1 1 1 1
    1 1         1 1 1
    1 1
  1 1 1       1 1 1
  1 1 1 1 1 1 1 1 1 1
  1 1 1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1 1 1 1
1 1 1 1 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)

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

(b)

Figure 3. A magnified version (by Algorithm 1) of the picture in (a) above is shown in (b) where the factor of magnification is 1.6. The area of the input picture is 120. So, the area of the output picture is theoretically supposed to be 307. But this area is in fact 302. The error (about 2%) is due to the digital nature of the picture.

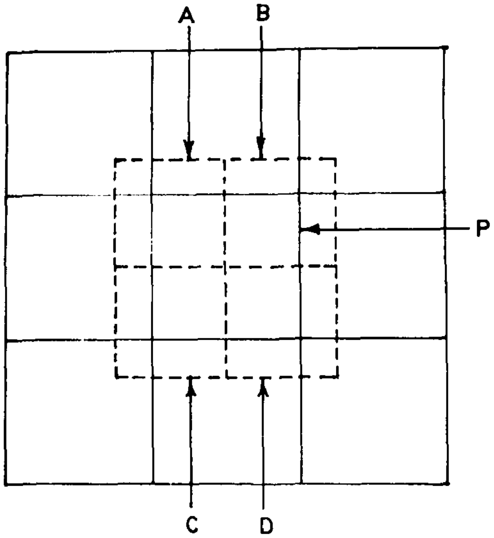

Figure 4. Above there are 9 input pixels (indicated by unbroken lines) among which P is the only 0-pixel. A, B, C and D (indicated by broken lines) are output pixels. The X value of each of these 4 pixels is 5/16. Thus, A, B, C and D are 1-pixels. So, the magnified picture has no hole though the input picture has one. This is possible because the input hole has only one 0-pixel and because the factor of magnification (4/3 here) is close to unity.

Property 1.2. *If a hole in a binary picture has only one 0-pixel, then after magnification the hole does not disappear, that is, gives rise to at least one output 0-pixel and the resulting output 0-pixels (if more than one in number) are 4-connected, provided the factor of magnification is more than $\sqrt{2}$.*

Property 1.3. *If a hole in a binary picture consists of only three 0-pixels (L-shaped) which fall neither in a single row nor in a single column, then after magnification with arbitrary factors the hole does not disappear and the resulting output 0-pixels are 4-connected.*

From Properties 1.1 and 1.3, follow the Properties 1.4 and 1.5 below.

Property 1.4. *Disconnected components of a binary picture cannot get connected after magnification with arbitrary factors.*

Property 1.5. *A hole in a binary picture does not disappear and remains 4-connected after magnification with arbitrary factors, provided the hole has more than one 0-pixel.*

Property 1.6. *A connected component with arbitrary thinness remains connected (with the same order of connectivity) after magnification if the factor of magnification is greater than or equal to 2.*

But if the factor of magnification is less than 2, a sufficiently thin connected component may get disconnected after magnification. One such example is given in Figure 5.

Property 1.7. *A connected component with thinness more than 1 remains connected (with the same order of connectivity) after magnification with arbitrary factors.*

From the above follows the property below:

Property 1.8. *If a binary picture is such that each of its components has thinness more than 1 and none of its components has a hole with only one 0-pixel, then the topological properties (i.e. surroundedness, connectedness, order of connectivity etc.) of the input picture do not change after magnification with arbitrary factors* (Parui, 1988).

3. Rotation

It is often necessary for shape matching to normalize the orientation of an object. We describe below a simple algorithm for rotating an object in two dimensions by arbitrary angles. Shape preserving porperties of rotation by this algorithm are then described.

Algorithm 2. To rotate a binary picture in anticlockwise direction by an arbitrary angle θ.

Let the input and output picture matrices be M and N respectively. Let M be an $m \times m$ matrix and N an $n \times n$ matrix. Since N has to accommodate all possible orientations of M, n has to be at least $\sqrt{2}\, m$. We take $n = [\sqrt{2}\, m] + 1$. Let $a = \cos\theta$ and $b = \sin\theta$.

Step 1. For a pixel (I, J) in the output matrix N, let $x = (I - n/2)a + (J - n/2)b$ and $y = -(I - n/2)b +$

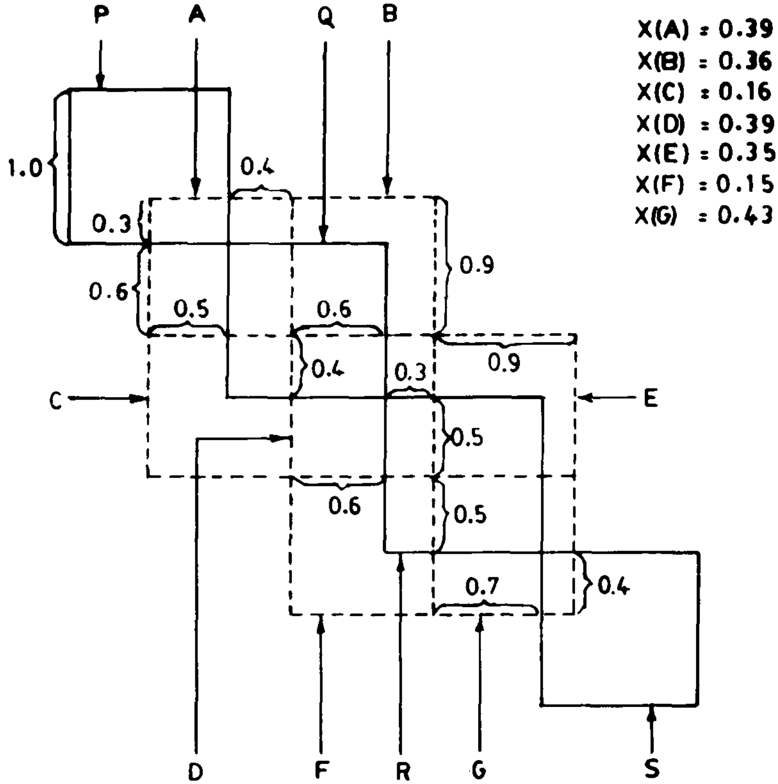

Figure 5. Above P, Q, R and S are input 1-pixels (indicated by unbroken lines) while A, B, C, D, E, F and G are output pixels (indicated by broken lines). The factor of magnification here is 10/9. From the X values above, it is clear that A, B, C, D, E and F are 0-pixels. Thus, a connected picture with thinness 1 (corner connected) can be disconnected after magnification.

$(J-n/2)a$. Note that (x, y) is the point obtained by rotating (I, J) around the centre of N by θ in the clockwise direction. Let I' and J' be the nearest integers of $x+m/2$ and $y+m/2$ respectively. If (I', J') falls outside the range of the input matrix M, set $N(I, J)=0$. Otherwise, set $N(I, J)=0$ if $M(I', J')=0$ and $N(I, J)=1$ if $M(I', J')=1$.

Perform Step 1 for all output pixels of N.

Note that in Algorithm 2 the output pixels are mapped through rotation to input pixels and thus each output pixel is assigned a value (0 or 1). Thus the output matrix is well defined. This would not be true if the input pixels were rotated to the output pixels. This is because some output pixels in that case would be left out.

The shape preserving properties of rotation by Algorithm 2 are given below.

Property 2.1. *If a hole in a binary picture has thinness more than* 1, *then after rotation by arbitrary angles, the hole gives rise to at least two output* 0*-pixels and the resulting* 0*-pixels are* 4*-connected.*

If however a hole has thinness 1, then after rotation the hole may get disconnected. One such example is given in Figure 6. From the property above we can say:

Property 2.2. *Disconnected components of a binary picture cannot get connected after rotation by arbitrary angles, provided the thinness of the background is more than* 1.

But if the thinness of the background is 1, then disconnected components may get connected after rotation. One such example is given in Figure 7.

Property 2.3. *A connected component with thin-*

```
      (5,1) (5,2) (5,3) (5,4) (5,5)
      (6,1) (6,2)       (6,4) (6,5)
      (7,1)                   (7,5)
      (8,1) (8,2)       (8,4) (8,5)
      (9,1) (9,2) (9,3) (9,4) (9,5)
```

(a)

```
                  (0,7)
            (1,6) (1,7) (1,8)
      (2,5)       (2,7)       (2,9)
(3,4) (3,5) (3,6)       (3,8) (3,9) (3,10)
      (4,5) (4,6) (4,7)       (4,9)
            (5,6) (5,7) (5,8)
                  (6,7)
```

(b)

Figure 6. The picture in (b) is the rotated version (by Algorithm 2) of the picture in (a) by 45 degrees in the anticlockwise direction. Note that the rotated picture has 4 holes though the input picture has only one. This is possible because the thinness of the hole in the input picture is as small as 1. (In Figures 6 and 7, 1-pixels are shown by their matrix coordinates and 0-pixels are suppressed.)

```
      (5,1) (5,2) (5,3) (5,4) (5,5)
      (6,1) (6,2) (6,3) (6,4) (6,5)

      (8,1) (8,2) (8,3) (8,4) (8,5)
      (9,1) (9,2) (9,3) (9,4) (9,5)
```

(a)

```
                  (0,7)
            (1,6) (1,7) (1,8)
      (2,5) (2,6) (2,7)       (2,9)
(3,4) (3,5) (3,6)       (3,8) (3,9) (3,10)
      (4,5)       (4,7) (4,8) (4,9)
            (5,6) (5,7) (5,8)
                  (6,7)
```

(b)

Figure 7. The picture in (b) is the rotated version (by Algorithm 2) of the picture in (a) by 45 degrees in the anticlockwise direction. Note that the input picture has 2 disjoint connected components which become one connected component in the output. This is possible because the thinness of the background (which is 1) of the input picture is sufficiently small.

ness more than 1 *remains connected* (*with the same order of connectivity*) *after rotation by arbitrary angles.*

From the above follows the property below:

Property 2.4. *If a binary picture is such that* (i) *its background has thinness more than* 1, (ii) *each of its components has thinness more than* 1 *and* (iii) *none of its components has a hole with thinness less than* 2, *then the topological properties* (*e.g., surroundedness, connectedness, order of connectivity etc.*) *of the input picture do not change after rotation by arbitrary angles* (Parui, 1988).

4. Conclusions

Two algorithms for magnification and rotation have been proposed and their shape preserving properties described. They may be used to construct a shape similarity measure for 2-dimensional regions (Parui, 1984) on the basis of which 2-dimensional shape discrimination is possible.

It can be seen that the above magnification algorithm in 2 dimensions can easily be extended to 3 dimensions. For this purpose, voxels instead of pixels are to be considered. In 2-dimensional magnification, each output pixel can intersect with at most 4 input pixels. Correspondingly, in 3 dimensions, each output voxel will have intersection with at most 8 voxels. The weights for these voxels can be computed in the same way as the weights (R_1, R_2, S_1 and S_2) for the 4 pixels in 2 dimensions are computed. Similarly, the X values and the output voxels N can be computed in the same manner. Rotation of a voxel will involve 2 angles in 3 dimensions and will again be similar to rotation of a pixel (involving only one angle) in 2 dimensions. The shape preserving properties of magnification and rotation in 3 dimensions will be quite similar to those (in 2 dimensions) discussed in the present paper. The 3-dimensional magnification and rotation algorithms can be useful in 3-dimensional shape matching (Parui and Banerjee, 1988).

Acknowledgements

The author wishes to thank the anonymous referee for helpful comments on the original manuscript.

References

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

Parui, S.K. (1988). Shape preserving properties of some operations on binary pictures. *Proc. 9th Intl. Conf. on Pattern Recognition*, Rome, 773-775.

Parui, S.K. and D.K. Banerjee (1988). Some operations on 3-D binary images for shape matching. *Proc. IEEE Intl. Conf. on Systems, Man and Cybernetics*, China, 1031-1034.

Parui, S.K. and D. Dutta Majumder (1984). How to quantify shape distance for 2-dimensional regions. *Proc. 7th Intl. Conf. on Pattern Recognition*, Montreal, 72-74.

Rosenfeld, A. (1970). Connectivity in digital pictures. *J. ACM* 17, 146-160.

Rosenfeld, A. and E.G. Johnston (1970). Geometrical operations on digitized pictures. In: B.S. Lipkin and A. Rosenfeld, Eds., *Picture Processing and Psychopictorics*. Academic Press, New York, 217-240.