

Shape Representation

Multiresolution Representation of Shape in Binary Images

Gunilla Borgefors¹, Giuliana Ramella², Gabriella Sanniti di Baja²

¹Centre for Image Analysis, Swedish University of Agricultural Sciences, Uppsala, Sweden
Fax: +46 18 553447; email: gunilla@cb.uu.se

²Istituto di Cibernetica, Italian National Research Council, Arco Felice, Naples, Italy
Fax: + 39 81 5267654; email: gr,gsdb@imagn.na.cnr.it

Abstract Binary pyramids can be used for multiresolution pattern representation. The two "standard" pyramids schemes, OR- and AND-pyramids, have, however, serious drawbacks, as they distort the shape significantly. In OR-pyramids black pixels spread all over the array due to expansion and merging of close regions. The shape of the original pattern is rapidly blurred. In AND-pyramids narrow regions of the initial pattern may either completely vanish or become disconnected. In both cases, the shape of the pattern is not preserved. Here alternative approaches, aimed at improving shape and topology preservation in binary pyramids, are presented. The first approach combines the OR and AND rules, whereas the second approach uses grey-level images as an intermediate step. The algorithms are easy to implement and produce significantly better results than the ones obtained by OR/AND pyramids.

Keywords: Binary pyramid, multiresolution representation, shape preservation

1. Introduction

Various representation schemes are possible for a discrete pattern in a binary image, [1]. Selecting the preferred scheme usually depends on the task to be accomplished. For many practical pattern recognition applications, multiresolution representations are preferred. They provide a flexible tool that has often been found to fit the user's needs better than single resolution systems.

Multiresolution representations can be obtained in two ways. Either we first convert the image to some representation of the contents and then manipulate the representation to create a multiresolution structure. Or we convert the image to a resolution pyramid and then use the pyramid to extract the (multiresolution) representation. The first approach includes, for example multiresolution skeletonization, whereas the second includes quadtree coding and the more and more popular scale space theory.

Multiresolution approaches to binary shape represent unique challenges, and have so far not been investigated thoroughly. In this paper we present several new methods for building good *binary* resolution pyramids, that can then be used for achieving good multiresolution shape representations.

The most straightforward example of a multiresolution image is the 2×2 binary pyramid, [2]. The highest, original, resolution level in a binary pyramid is a bitmap representation, where pixels are either black (pattern) or white (background). The next,

lower, resolution level is built by partitioning the bitmap in 2×2 blocks of pixels, the "children," and associating a single pixel, the "parent," to each block. The colour of the new pixel is determined according to a fixed *rule*. That four black (white) children should have a black (white) parent is evident. The rule is needed to treat the "mixed" cases. The process is repeated for the previously computed lower resolution representation, and then further iterated to build all possible resolution levels, ending in a single pixel in the lowest resolution level. (If the original image is not $2^n \times 2^n$, white rows and columns are added as needed.)

The logical OR and AND operations are the two most commonly used rules when building binary pyramids. They are easy to implement and it is easy to predict and understand the results. However, when OR-pyramids are built, the black pixels spread all over the array due to the large number of 2×2 configurations that contain a black pixel and thus have a black parent. As a result, the pattern is expanded, regions of the pattern sufficiently close to each other merge, and the shape of the original pattern is rapidly blurred. On the other hand, when AND-pyramids are built, the pattern is shrunk and narrow regions of the initial pattern may either completely vanish or become disconnected as the resolution decreases. This is because only one 2×2 configuration has a black parent. In both cases, the shape of the pattern is not adequately preserved. To preserve pattern shape better, something in between the AND and OR pyramids is desirable.

In addition to the rule used to determine the colour of the pixels at lower resolutions the *position* of the grid used to partition the array into 2×2 blocks influences the shape of the resulting pattern. This problem is seldom discussed, probably because it is much less pronounced in grey-level imagery (even if it is present there too). The grid can, in fact, be shifted in four different positions over the image, thus originating four differently shaped patterns at the next resolution level. Some of these patterns resemble the original shape better than others, but there is no way to select the best grid position a priori. Using a combination of the four possible resulting images could produce more stable representations.

We have recently addressed the task of preserving shape in binary pyramid representation, [3]. In this paper, we continue in the same direction and present further approaches to improve shape preservation. Our methods are designed to preserve perceived shape, more than to maintain topology. However, topology *is* fairly well preserved by our methods. Changes of binary topology are, in any case, unavoidable when image resolution is decreased. Maintaining the connectedness in the pattern and avoiding the creation of holes (i.e., maintaining the connectedness of the background) are dual problems that can never be solved simultaneously. Our approaches favour maintenance of pattern connectedness.

2. Combinations of Binary Images

Let I_1 be the $2^n \times 2^n$ original black and white image, stored in the bottom level (called the first level) of the pyramid. Each of the successive n levels of the pyramid (I_2, I_3, \dots, I_{n+1}) is built from its preceding level. Every pixel of level I_k , $1 < k \leq n-2$, is either black or white, depending on the colours of its four children in I_{k-1} . Since the last three levels consist of 1, 4, and 16 pixels, respectively, and hence are not meaningful for shape representation, we will define in the following the resulting pyramid as a structure consisting of $n-2$ levels, where the 8×8 level is the last one.

All our methods will be illustrated on the same small ($2^6 \times 2^6$) test image shown in Fig. 1. This $2^6 \times 2^6$ image is the first level of the pyramid. As this level is always the same, whatever scheme is used to build the pyramid, we will in the following show only the three lower levels, consisting of $2^5 \times 2^5$, $2^4 \times 2^4$, and $2^3 \times 2^3$ pixels, respectively. Using a larger image as a test example is not necessary, as shape and topology preservation problems are more prominent at low resolutions.

We first compute the commonly used OR- and AND pyramids, where the logical OR and AND operations are employed as building rules. To compute level I_k of the OR-pyramid (AND-pyramid), all pixels with at least one black child (four black children) in I_{k-1} become black. In Fig. 2, the OR-pyramid and the AND-pyramid of the test image are shown. It is easy to see that in both cases shape and topology are not preserved at all.

The logic rules can be interpreted as "arithmetic" rules, where the sum of the values of the four children, Σ_p , is computed. In the OR- and the AND- pyramids a parent becomes black if $\Sigma_p \geq 1$ and $\Sigma_p = 4$, respectively. Other arithmetic rules could be used, based on Σ_p . Two rather obvious pyramid rules, that can be seen as intermediate

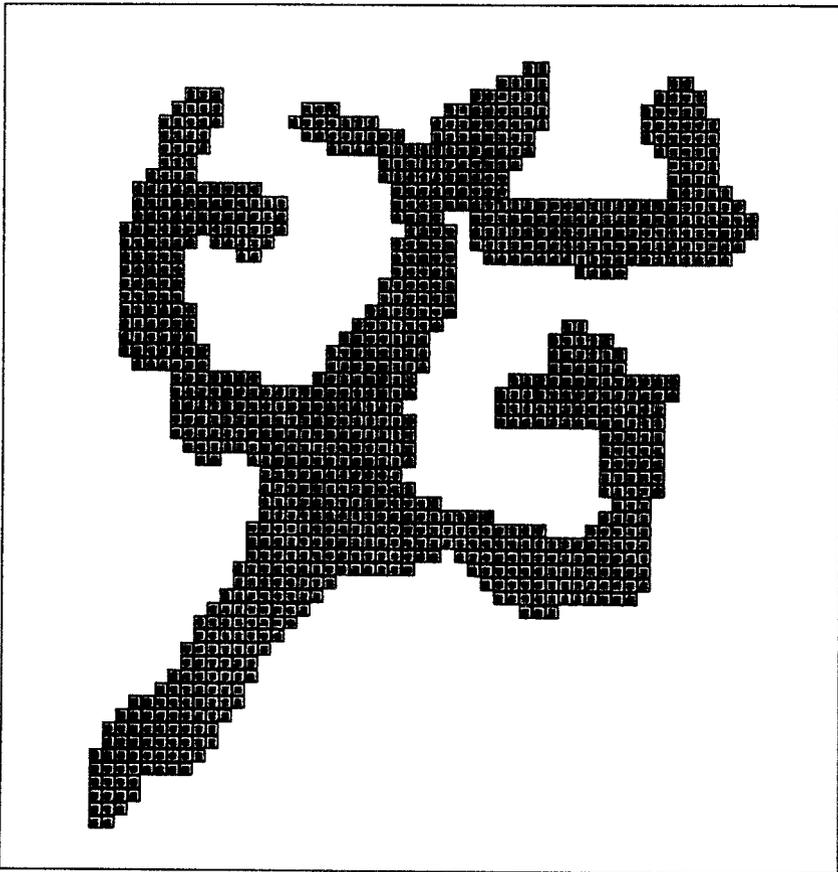


Fig. 1. First level, I_1 , of a binary pyramid, where a $2^6 \times 2^6$ test image is stored.

between the OR and the AND rules, are obtained by setting to black pixels with $\Sigma_p \geq 2$ and $\Sigma_p \geq 3$, respectively. Although the shapes of the resulting patterns are better than those provided by the OR- and the AND-pyramid, shape and topology are still barely preserved.

To get better, more stable, pyramid representations we use, at each resolution level, combinations of the four images that can be obtained in correspondence to the four different positions of the partitioning grid. The pyramid rules used are the simple arithmetic rules just described. Let $A_2, B_2, C_2,$ and D_2 be the four images, obtained by shifting the grid used to partition I_1 into 2×2 blocks. The rule $\Sigma_p \geq i$ is used to build $A_2, B_2, C_2,$ and D_2 . When combining these four images, the sum of the four pixels values in the same position, Σ_g , is computed for all pixels. The combined level I_2 is obtained setting to black all pixels with $\Sigma_g \geq j$. Similarly, $A_3, B_3, C_3,$ and D_3 are obtained by shifting the grid over I_2 , and are combined to build level I_3 . The process is iterated until all resolution levels are computed.

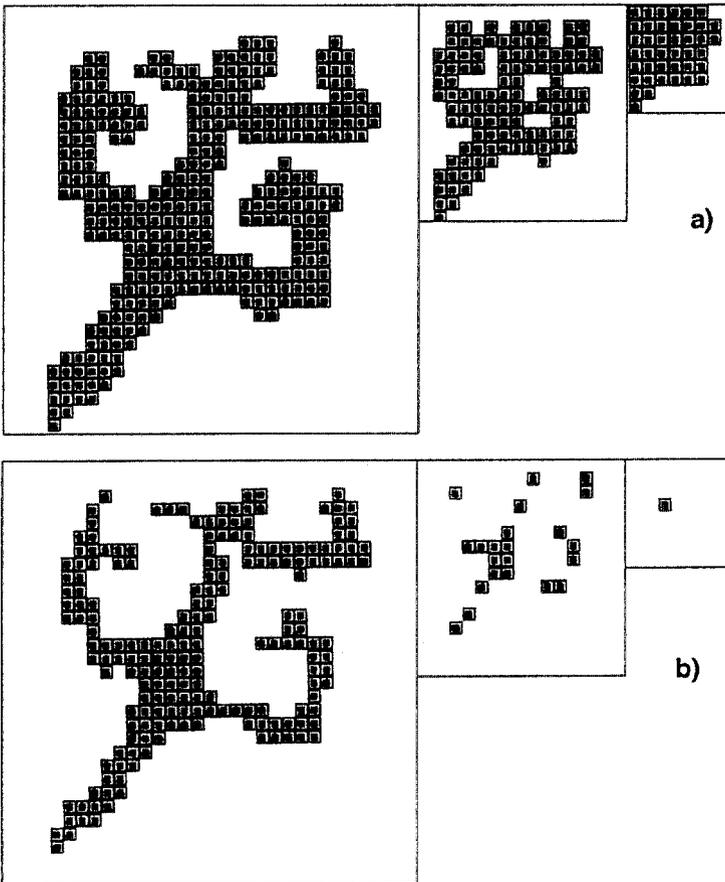


Fig. 2. Levels I_2, I_3 and I_4 of the OR-pyramid, a), and the AND-pyramid, b), are shown from left to right.

Pyramid rule and image combinations giving good results are the following:

- $\Sigma_p \geq 1, \Sigma_g = 4$ (equivalent to AND-ing four OR images)
- $\Sigma_p = 4, \Sigma_g \geq 1$ (equivalent to OR-ing four AND images)
- $\Sigma_p \geq 2, \Sigma_g \geq 3$ (using intermediate rules)
- $\Sigma_p \geq 3, \Sigma_g \geq 2$ (using intermediate rules)

Using all the four images $A_2, B_2, C_2,$ and D_2 makes the representation more stable. Using two different operations to build the images and to combine them, improves shape preservation.

The best results are obtained when using the intermediate rules, i.e. the existence of first two and then three (or three and two) black pixels. The corresponding results computed on the test pattern are shown in Fig. 3. The two pyramids differ slightly from each other, but in both cases they are significantly better, concerning shape preservation, than those achieved when simply building the pyramids as shown in Fig. 2, especially if we consider the "worst" grid placement.

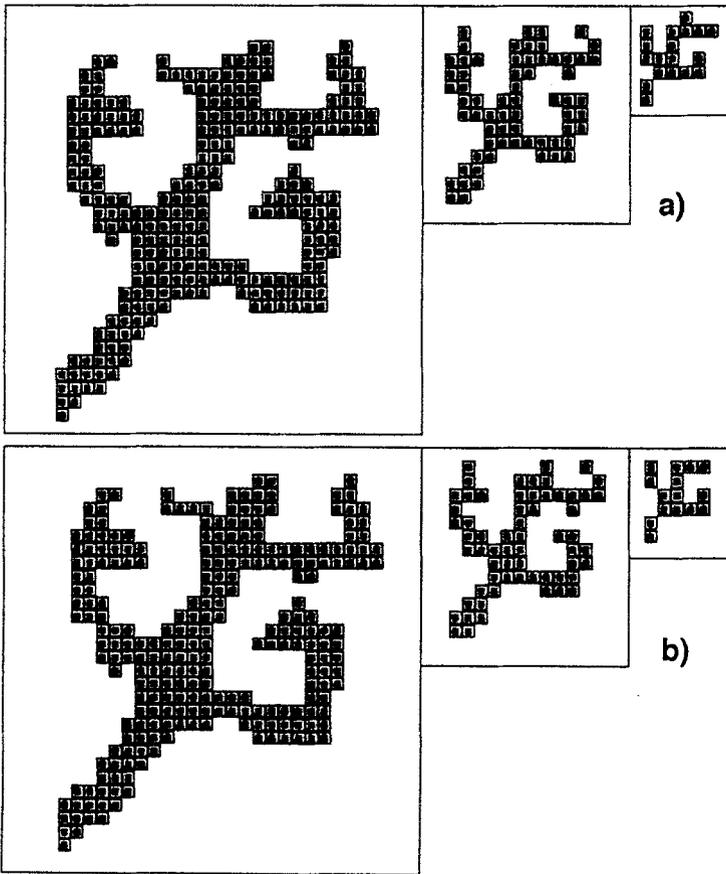


Fig. 3. Pyramids obtained by combining the four images obtained at different grid placements, using intermediate rules; $\Sigma_p \geq 2, \Sigma_g \geq 3$ a); $\Sigma_p \geq 3, \Sigma_g \geq 2$ b) (see the text).

It should be pointed out that although these "combined" pyramids are much less sensitive to the placement of the 2×2 grid, the results are still not translation independent. The reason is that when the grid is shifted a new column (row) of white pixels must be added and this column (row) can be added either at the top (left) or at the bottom (right) of the image, giving again four possibilities.

3. Using Intermediate Grey-Level Images

A combination of the images that would be obtained in correspondence with the four different positions of the partition grid can be achieved without actually building the four images. By weighting the neighbours of the pixels in the first level of the pyramid, using appropriate weights, a grey-valued intermediate second level image is obtained. This single image contains the same information as the four shifted binary images. After this grey-valued image has been binarized according to some rule, the second binary level is completed. The process is iterated until all levels have been completed.

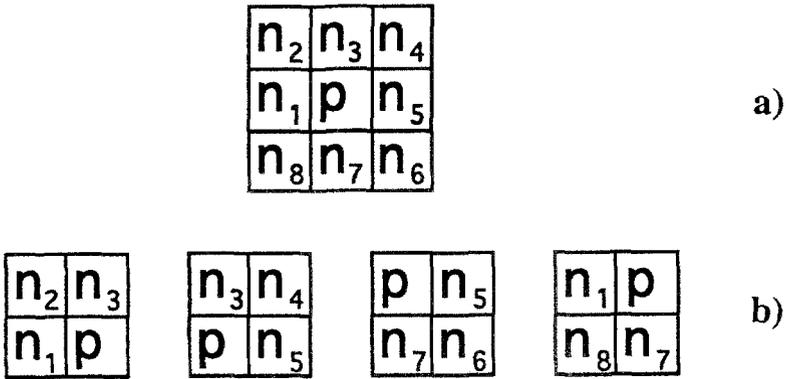


Fig. 4. A 3×3 window centred on a pixel p , a); and the four 2×2 blocks including p that can be obtained by shifting the partition grid, b).

Let n_1, n_2, \dots, n_8 denote the eight neighbours of any pixel p , see Fig. 4a. In particular, suppose that p is a pixel in the bottom, first, level of the pyramid, having even co-ordinates (i.e., p belongs to row $2i$ and column $2j$). Depending on the position of the grid on the first level image, the pixel p belongs to one of the four 2×2 blocks shown in Fig. 4b. Each of these blocks would generate one pixel (either black or white) on one of the four next level binary images. To build the second level of the pyramid by taking into account the value that p would have in all four images, we use the 3×3 multiplicative mask illustrated in Fig. 5. In this way, a grey-valued image G_k

1	2	1
2	4	2
1	2	1

Fig. 5. Mask of weights for computing the intermediate grey-valued image.

is built where the possible value of p ranges from 0 (when p and all its neighbours are white) to 16 (when p and all its neighbours are black).

Binarization of the grey-valued image G_k can be accomplished by following different criteria. Generally, pixels with a small values (close to 0) can be safely set to white, as their contribution to the next level of the binary pyramid is disregardable whatever the position of the grid happen to be. Analogously, pixels with large values (close to 16) can be set to black, as they would probably be set to black in almost all the four images obtained by shifting the grid. For the remaining pixels, the final colour is decided by taking into account the value of the pixels as well as those of their neighbours. Note that thresholding done by setting all pixels with grey-level $g > 0$ or $g = 16$ to black does *not* produce the OR and the AND images, as the grey-level image represents all four shifted images.

We have tested many criteria for the binarization of the grey-valued image. The two best criteria for shape preservation that we have found are the following. Let a pixel p have value g .

Criterion 1:

If $0 \leq g \leq 4$ set p to white.

If $5 \leq g \leq 10$ set p to black if it is locally maximal along either the horizontal or the vertical direction, otherwise set it to white.

If $11 \leq g \leq 16$ set p to black.

The rule for $5 \leq g \leq 10$ is intended to preserve pattern connectedness while avoiding unnecessary thickening.

Criterion 2:

If $0 \leq g \leq 5$ set p to white.

If $6 \leq g \leq 10$ set p to black if g is larger than the average value in a 3×3 neighbourhood centred on p .

If $11 \leq g \leq 16$ set p to black.

The rule for $6 \leq g \leq 10$ is intended to preserve the most "significant" out of several neighbouring pixels.

The results computed from the test pattern by applying above criteria to the grey-valued intermediate images the are shown in Fig. 6. Slight changes of the thresholds will not give drastically different results, but can be used to fine-tune the pyramid for different applications.

4. Conclusions

A number of methods have been presented to improve shape preservation when representing patterns in binary pyramids. They are all easy to implement, and produce significantly better results than the ones obtained by the standard OR/AND pyramids. They are also better for topology preservation, but remember that topology can not be accurately preserved in decreasing resolutions no matter what you do. This is, in fact, the reason why we concentrate on shape preservation rather than topology.

Our first approach uses only binary images. It builds a binary pyramid by using suitable combinations of the four images obtained by shifting the partition grid. In these four images the value of a pixel depends only on the number of its black children. In the combined image the value of a pixel depends on the number of times the pixel is black in the four images. We found that requiring at least 2 (3) black children and at least 3 (2) black pixels in the four images gave the best results.

Our second approach replaces the four binary images with a single grey-level image. This image is computed by a 3×3 mask of weights, which take into account the four 2×2 configurations that can be placed within the 3×3 neighbourhood of a pixel. This grey-valued image is binarized according to one of two rules, to obtain the next binary level. The rules do not only consider the value of a single pixel, but also those of its neighbours.

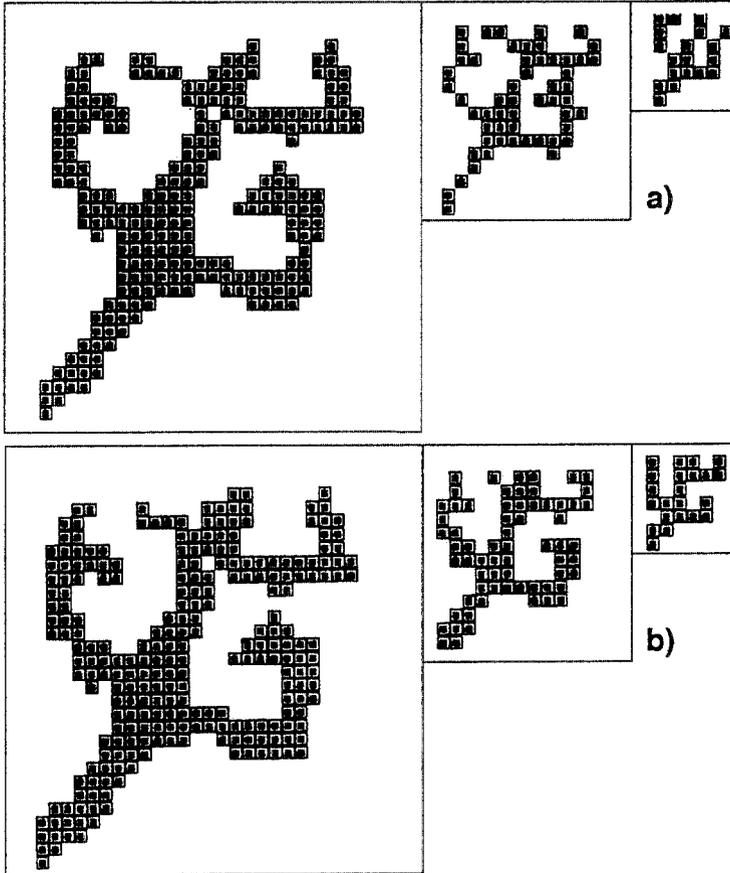


Fig.6. The two pyramids obtained using intermediate grey-valued levels. A pixel p is set to black if it is locally maximal along either the horizontal or vertical direction, a). A pixel p is set to black if its value is larger than the average value in its 3×3 neighbourhood, b).

References

- 1 A. Rosenfeld, A.C. Kak, *Digital Picture Processing*, Academic Press, Orlando, FL, 1982.
- 2 H. Samet, *Applications of Spatial Data Structures*, Addison-Wesley, Reading, MA, 1989.
- 3 G. Borgfors, G. Sanniti di Baja, "Shape Preserving Binary Pyramids," *Proc. Portuguese Conference on Pattern Recognition (RECPAD'96)*, Guimarães, Portugal, 197-203, 1996.