**UNIVERSITAT POLITÈCNICA DE CATALUNYA**

*Departament de Teoria del senyal i comunicacions*

# MULTIRESOLUTION IMAGE SEGMENTATION BASED ON COMPOUND RANDOM FIELDS: APPLICATION TO IMAGE CODING

Autor: Ferran Marqués
Director: Antoni Gasull

Barcelona, Diciembre de 1992

# CHAPTER V

# SEED EXTRACTION BY MORPHOLOGICAL TOOLS

As stated in the previous chapters, both the monoresolution and the multiresolution segmentation approaches fail when coping with interior regions. An interior region has been defined as a region such that its contour does not touch the contours of other regions at any point. The reason for this failure is owing to the procedure for selecting, from an initial segmentation, new possible states (see Section III.2). This procedure is based on the label change of pixels laying on the boundaries of current regions. That is, a new state is reached by giving to a pixel the label of one of its neighbours. Therefore, if an interior region has been overlooked in the initial segmentation, the labels of its pixels will not change throughout the whole procedure. This drawback is mitigated by the use of multiresolution approaches, given that their global analysis leads to better initial segmentations. Nevertheless, even when such approaches are used, some interior regions are not detected.

## V.1.- Necessity of using seeds

The first trial to overcome this drawback relies on varying the procedure for selecting new states. The new procedure of selection should allow pixels of interior regions to change their label. That is, labels of pixels laying on the interior of the current regions should be checked. In this way, the possibility of directly creating new regions is introduced in the algorithm. Therefore, new states are reached by giving to a pixel the label of any of its neighbours or a new label different from the current labels in the partition.

### V.1.1.- Creation of new regions

The direct application of this procedure results in several problems. Note that new regions, which only contain a single pixel, can be created. As it has been emphasised in precedent chapters, parameter estimations performed on small amounts of data are not reliable. In this case, the variance estimation for the new region will be zero. Such a value is not allowed for computing the conditional probabilities as proposed in (III.9). This problem can be circumvented by assuming a very low, positive value as variance of one pixel regions. Nevertheless, this solution implies to introduce an ad hoc value in the procedure.

A second problem appears in relation to textured areas. The basic monoresolution algorithm presented in Chapter III performs a local analysis in order to maximise the joint probability. The fact of being local prevents for correctly coping with textured areas. Hence, the necessity of using multiresolution approaches for segmenting such areas (see Chapter IV). In the multiresolution approach, local analyses at different resolutions are only allowed relying on the information provided by the coarsest levels. That is, fine level segmentations are mainly based on coarse level contour refinements. However, if the algorithm is allowed to carry out a local analysis at every point of each level, the global viewpoint supplied by the multiresolution approach is useless. This kind of local analyses leads to oversegmented textured areas, since small homogeneous zones forming the textured are detected as single regions.

In addition, checking whether every point within the image may form a new region or not can double the computational load of the algorithm. For instance, an image as the Cameraman contains, at most, 23.000 pixels laying on its boundaries (this is the maximum number found in the segmentations

presented in Chapter III and Chapter IV), while the whole image contains 65536 pixels.

These effects can be illustrated by the segmentations of the Cameraman image presented in Figure V.1. The first segmentation (236 regions) has been obtained by using the unsupervised approach presented in Chapter IV (actually, it is the segmentation of Figure IV.19). On its turn, the second segmentation (1024 regions) has been achieved by the same algorithm, but allowing the direct creation of new regions. In this algorithm, once a new region has been created, its parameters are computed and introduced in the image model. Therefore, this new region can be expanded by the algorithm. That is, a single-pixel new region can act as a seed for the whole interior region to be detected.

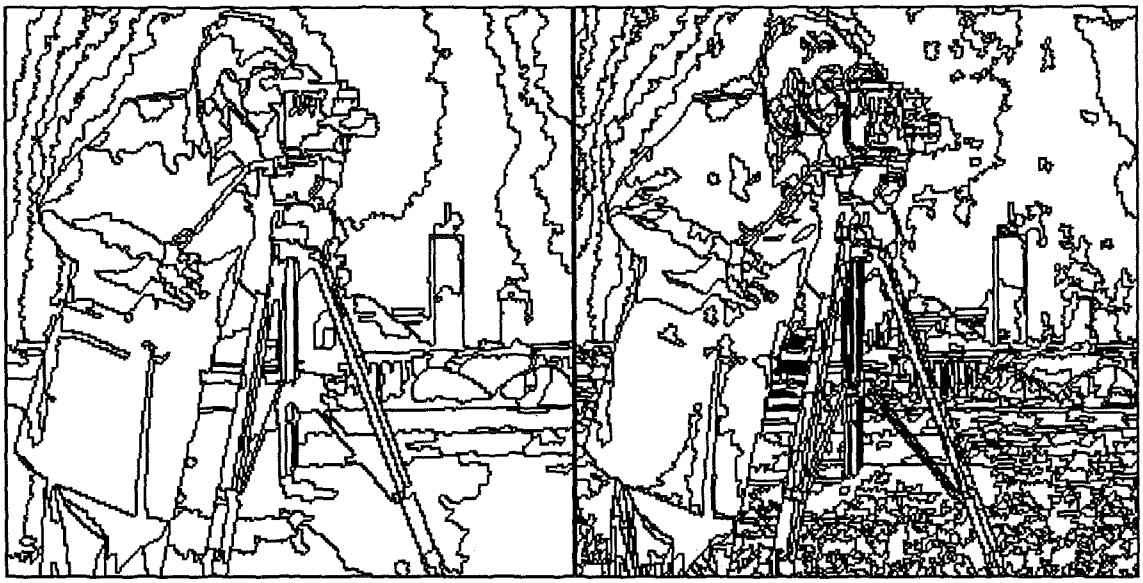

**Fig. V.1.- Effect of allowing the direct creation of new regions**

As it can be seen in Figure V.1, several details, overlooked in the first segmentation, appear in the second one (e. g.: the right stick of the tripod or some details in the face of the man and in the camera). However, some of these details, which should be detected as single regions, have been oversegmented (e.g.: the right stick of the tripod or details in the camera). This oversegmentation is owing to the fact that more than one single-pixel region has been created in the area corresponding to a unique interior region. The reason for having this multiple seed effect is owing to the scanning of the image as well as to the shape of interior regions. For instance, an interior region having a U-like shape may yield two different new regions with a

normal raster scanning. In addition, textured areas appear clearly oversegmented in the second image (e.g.: the grass). Furthermore, the computational load of the second segmentation is 2.4 times that of the first case.

Two main ideas result from this example. First, the possibility of allowing the algorithm to directly create new regions can solve the problem of detecting interior regions. However, new regions must not be create at any location within the image, in order not to spoil the multiresolution approach improvements. Thus, some external information should guide the algorithm to the locations where new regions can be created, while forbidding the creation in other places. That is, the algorithm should be provided with a set of seeds aiming at the points at which creation of new regions is feasible. The information contained on this set of seeds can be represented by a binary image.

In addition, some mechanism should be introduced in the algorithm to prevent the creation of several small regions in a zone which could be gathered in a single one. Thus, in the optimum case, only one seed should mark the possible existence of a single interior region.

### V.1.2.- The error image

To analyse the presence of non-detected interior regions within an image, given a segmentation, the error image is built. This image contains the difference between the original image and the mosaic representation of the segmentation. It should be recalled that a mosaic image is a representation of the partition of an image with each region filled by its mean value. Therefore, the error image represents the difference between the image model and the original image, withdrawing the variance information. In a following section, the use of the variance information will be discussed.

The information stored in error images is similar to that contained in the levels of the Laplacian pyramid (see Chapter IV). Assuming that segmentations used for computing error images have been correctly performed (they have correctly located their regions), error image information should be mainly related to textures in the original image and non-detected interior regions. Actually, some contour information appears in the error image, as well. The reason for having this information is the zone of uncertainty present around all contours in an image (see Chapter I). However, to remove this information from the error image is an easy task since the location of contours

in the segmented image is known. Therefore, the main task when extracting the seeds from the error image is to discriminate between texture and non-detected interior region information.

Classical thresholding techniques cannot be used for such a goal. The information about the absolute value of pixels in the error image is not enough to discriminate between both classes of information. That is, textured areas may yield pixels in the error image with larger absolute values than those produced by non-detected interior regions. Furthermore, the number of seeds per region depends very much on the chosen threshold. Finally, the situation of having a non-detected interior region embedded in a low gradient region is not correctly solved by a simple thresholding.

To illustrate the above concepts, the error image of the Cameraman and the first segmentation of Figure V.1 is shown in Figure V.2. In order to avoid representing negative numbers, an offset has been added to the whole image. That is, zero differences are represented by a gray level value of 128 in the image. After adding the offset, the image has been clipped so that its pixel values can be represented with one byte.

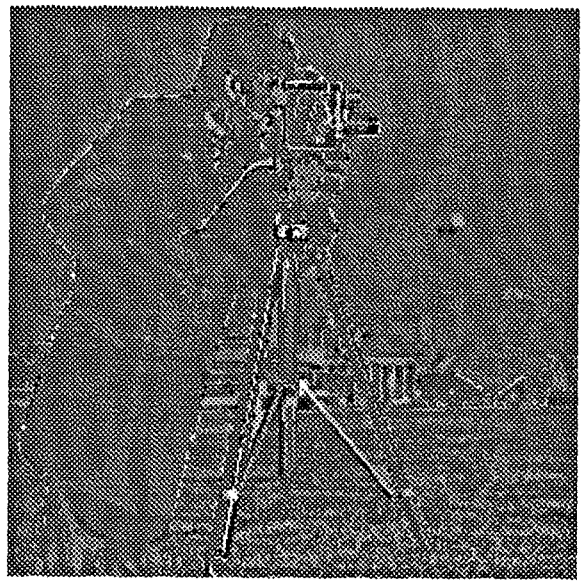

**Fig. V.2.- Example of error image**

Several details appear clearly marked in this error image; for instance, the right stick and the left leg of the tripod, some parts of the camera and the face of the man or the aerials of the large building. Note that contour information is also present in the error image. This information can be clearly

seen in the transitions between the region of the coat or the head and that of the sky. In addition, texture fluctuations appear due to the grass zone. It should be noticed that the absolute gray level values of these fluctuations can be quite high.

In order to detect overlooked interior regions, the segmentation method of Chapter IV may be applied to the error image. However, this approach arises some problems. Note that this segmentation cannot take advantage of the multiresolution analysis performed for segmenting the original image. That is, the segmentation of the error image produced at level (l) of the decomposition cannot be used to guide the segmentation of the error image at level (l-1). The reason for this statement is that, if the error image at a given level (l) is used to refine the final segmentations at this level (l), two consecutive error images should present very little similarity.

Therefore, a new multiresolution segmentation should be carried out, taking the error image at the bottom of the pyramid as basis of the decomposition. In this case, the computational load for the whole procedure is, at least, twice that of the method presented in Chapter IV. Furthermore, after performing both segmentations, information obtained in each one has to be gathered into a single result. This final step is not a simple task since some contours may be represented in both results. Given that segmentations have been carried out on different images with different model parameters (note that the Laplacian pyramid is different for each segmentation), related contours are not equal. Thus, a contour matching technique is necessary.

Simpler techniques for extracting seeds from non-detected interior regions should be sought. Image contrast turns out to be a useful information to determine whether a non-detected interior region may be present or not. In addition to this information, and towards the goal of distinguishing between textured zones and non-detected interior regions, homogeneity of contrasted areas can be used. That is, homogeneous contrasted areas are more likely related to interior regions than non-homogeneous one. Since morphological tools are well know to be very efficient for detecting contrasted, nearly flat regions, techniques relying on such tools have been studied.

## V.2.- Morphological tools

Mathematical Morphology (MM) is a very wide field which has multiple applications in the signal processing framework. This section is aimed at giving only a brief overview of the morphological tools used in this work, rather than an exhaustive study of the whole field. Furthermore, the analysis of such tools is mostly restricted to the scope of the current application; that is the detection of interior regions. For a complete discussion on the topic, the reader is referred to [14, 77].

The different operators discussed on this section can be defined as dealing with N dimensional signals. However, for simplicity, only the one and two dimensional cases are taken into account. Whichever the case, signals are denoted by $w_i$, $x_i$ and $y_i$. In addition, this study is only related to morphological operators using flat structuring elements. Flat structuring elements are very used since they lead to operators which preserve edges, commute with some modifications of the gray level scale and allow fast implementations [78]. In the sequel, a flat structuring element of size n is denoted by $S_n$.

In order to illustrate the properties of the different operators, the signal plotted on Figure V.3 is used. This signal has been generated following the behaviour of error images. That is, it has contrasted, nearly flat zones of both signs (similar to interior regions) and areas with rapid, contrasted fluctuations (similar to textured zones). In all the examples performed with this signal, a symmetric, flat structuring element of size 21 is used.

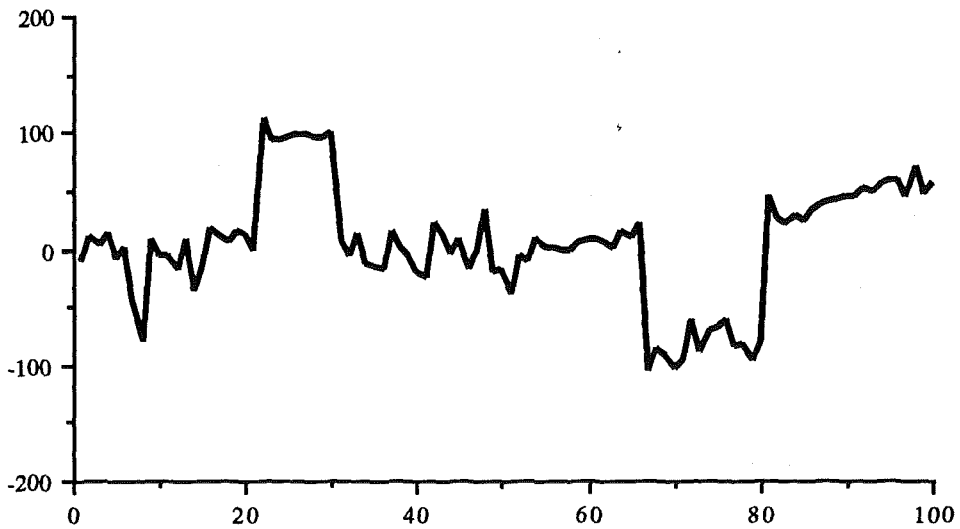


**Figure V.3.- Original monodimensional signal**

In the current application, a correct discrimination among the different kinds of information appearing in this signal should isolate both the positive and negative large peaks while removing the rapid fluctuations. In addition, peaks should be detected as single items. That is, a unique, connected zone should mark the presence of each peak. In this way, the possibility of introducing more than one seed per peak is avoided.

### V.2.1.- Erosion and dilation

Erosion and dilation are the basic operations of mathematical morphology and most of the morphological tools are based on them [14]. Their definition, in the case of discrete signals and flat structuring elements, can be given by:

$$y_i = \varepsilon_n (x_i) = \text{Min} \{ x_{i+k}, k \in S_n \} \quad (V.1)$$

$$y_i = \delta_n (x_i) = \text{Max} \{ x_{i-k}, k \in S_n \} \quad (V.2)$$

where $\varepsilon_n$ and $\delta_n$ stand for the erosion and the dilation performed with the structuring element $S_n$, respectively. The main properties of these operators are that they are increasing

$$\text{if} \quad w_i \subset x_i \quad \text{then} \quad \varepsilon (w_i) \subset \varepsilon (x_i) \quad \text{and} \quad \delta (w_i) \subset \delta (x_i) \quad (V.3)$$

and, if the centre of the structuring element is contained in the structuring element itself, they are anti-extensive and extensive respectively,

$$\varepsilon (x_i) \subset x_i \quad \text{and} \quad x_i \subset \delta (x_i) \, . \quad (V.4)$$

This last property can be observed in the example of Figure V.4.

Note that, given the anti-extensive (extensive) property, the eroded (dilated) signal is a kind of inferior (superior) envelope of the original signal. Moreover, erosion (dilation) expands the negative (positive) peaks of the signal, while shrinking, or even removing, the positive (negative) ones. Therefore, none of these operators is self-dual. That is, they handle in a different way positive and negative samples.

Although the eroded (dilated) signal is simpler than the original one and highlights the negative (positive) peaks, it cannot be used as detector for interior regions. Note that, after transforming, all peaks are either removed or

expanded. That is, these operators do not discriminate between textured areas and interior regions. On the contrary, they cope with both informations in a similar way. Furthermore, since negative (positive) peaks are expanded, their actual boundaries are lost. This effect does not allow to locate correctly the zones related to peaks. Therefore, more elaborated transforms have to be used.

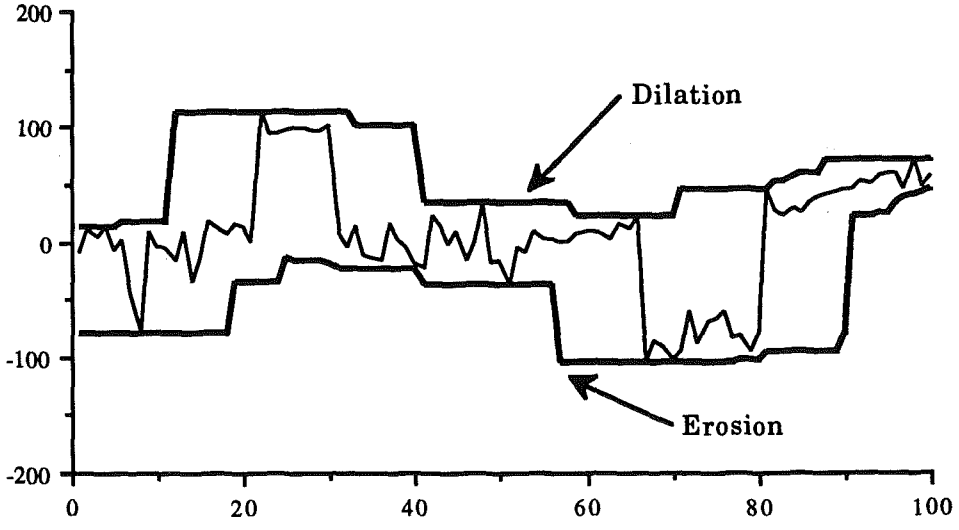


**Figure V.4.- Examples of erosion and dilation**

### V.2.2.- Open and close

Two transforms can be defined by composition of the operators above given. In this way, morphological open and close are defined as [14]:

$$\gamma_n = \delta_n \varepsilon_n \quad \text{and} \quad \varphi_n = \varepsilon_n \delta_n , \tag{V.5}$$

where $\gamma$ and $\varphi$ stand for open and close, respectively. Both transforms share the properties of being increasing and idempotent. This last propierty means that, if the transform is composed with itself (applied twice), the result is the same as if applied only once

$$\gamma = \gamma\gamma \qquad \varphi = \varphi\varphi . \tag{V.6}$$

Increasing and idempotent transforms are called morphological filters. Note that, besides these properties, open is anti-extensive while close is extensive. Therefore, both are not self-dual. The behaviour of these filters is illustrated in Figure V.5.

Since the open transform is anti-extensive (extensive), the opened (closed) signal is also an inferior (superior) envelope of the original signal. However, in this case, the filtered signal conforms better to the original signal and specially to its negative (positive) peaks. Note that, as commented above, edges are well preserved owing to the use of flat structuring elements. Therefore, it may be said that filtered signals are simplified versions of the original signal from which positive (negative) peaks not fitting within the structuring element have been removed.

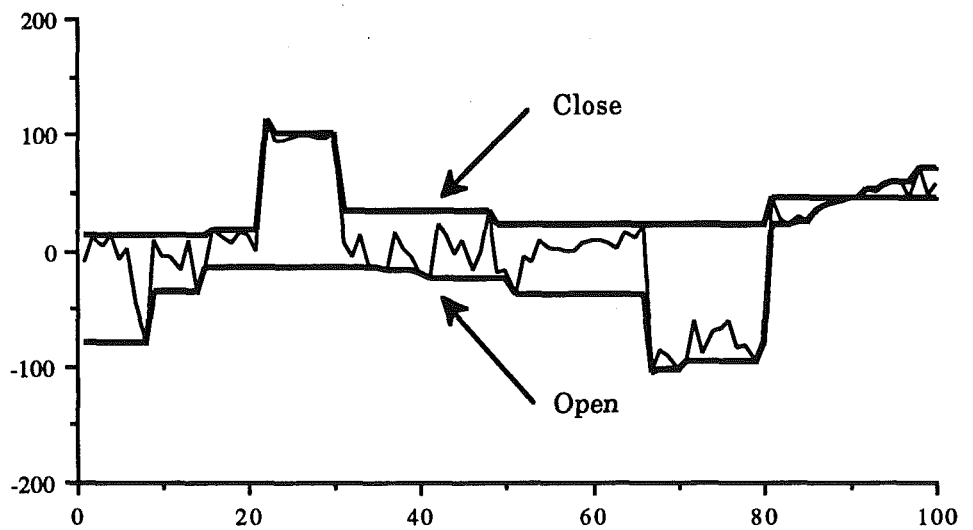


**Figure V.5.- Examples of open and close**

Note that, although peaks of the original signal have been highlighted and fluctuations have been removed after filtering, a detection of peaks carried out directly on the filtered signal is not possible. Peaks, rapid fluctuations and slow gradients may yield similar values, and, therefore, a basic detection (e.g.: thresholding) cannot be applied.

### V.2.3.- Residues with open and close

Another kind of transforms can be obtained by computing the difference between morphological filters. This kind of morphological transforms are the so-called residues. By computing the difference between the identity and an open or a close filter, two different residues are achieved. These residues are denoted by the name of white and black Top Hat, respectively [79]. These transforms are given by:

$$\begin{aligned} &\text{white Top Hat} = I - \gamma \quad (\Rightarrow y_i = x_i - \gamma(x_i)) \\ &\text{black Top Hat} = \varphi - I \quad (\Rightarrow y_i = \varphi(x_i) - x_i)\,, \end{aligned} \tag{V.6}$$

where I stands for the identity operator. These transforms are neither increasing nor extensive or anti-extensive but they are idempotent. The white (black) Top Hat isolates the elements in the original signal removed by the open (close) filter. An example of the application of the white Top Hat is plotted in the graphic of Figure V.6. In this case, the graphic shows the original signal, the opened signal and the result of the white Top Hat transform. In order to show these signals in the same graphic, an offset (300) has been subtracted from the Top Hat result.

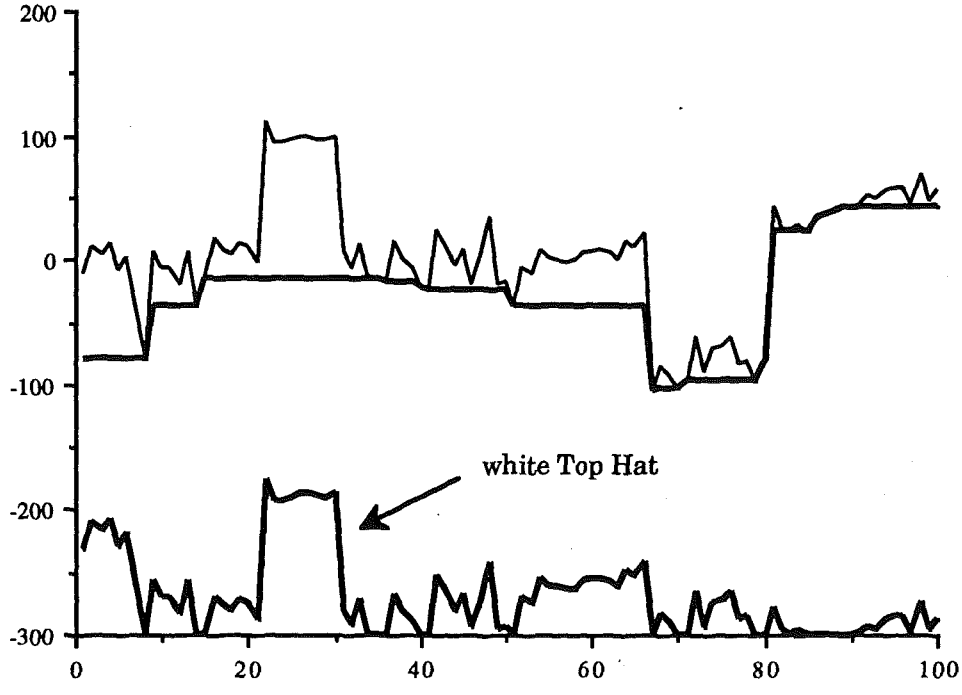


**Figure V.6.- Example of white Top Hat**

Slow variations have been removed from the original signal, given that the transformation is a residue of a filter following them. However, since the open (close) filter is anti-extensive (extensive), it yields an inferior (superior) envelope of the signal, which removes the rapid positive (negative) fluctuations. Therefore, these fluctuations appear in the resulting residue. On the other hand, it is worth noticing that white (black) Top Hat preserves the shape of positive (negative) peaks.

Although residues are a big step towards the detection of interior regions, white and black Top Hat do not provide with an efficient tool for such a task. The main problem results from the fact that rapid fluctuations (textures) are strongly represented in the final result. Moreover, since the filters are not self-dual, informations obtained from both transformations has to be merged in a single result. This merging procedure may lead to some problems. Hence, the necessity of using other kind of filters as basis for the residue transformation.

### V.2.4.- Close_open and open_close

By composing open and close filters, a few new filters can be created [80]. These filters, in spite of not being self-dual, cope with both positive and negative samples in a more similar way than the basic open and close filters. The definition of these filters is as follows:

$$\begin{aligned} &\text{close\_open} = \varphi\,\gamma && \text{open\_close\_open} = \gamma\,\varphi\,\gamma \\ &\text{open\_close} = \gamma\,\varphi && \text{close\_open\_close} = \varphi\,\gamma\,\varphi\,. \end{aligned} \qquad \text{(V.7)}$$

Note that, due to the idempotence property, these are all the filters which can be built by composition of open and close. As filters, they are idempotent and increasing. However, in this case, they are neither extensive nor anti-extensive. This is the reason why they are closer to being self-dual than open and close filters. In Figure V.7, examples of close_open and open_close filterings are shown.

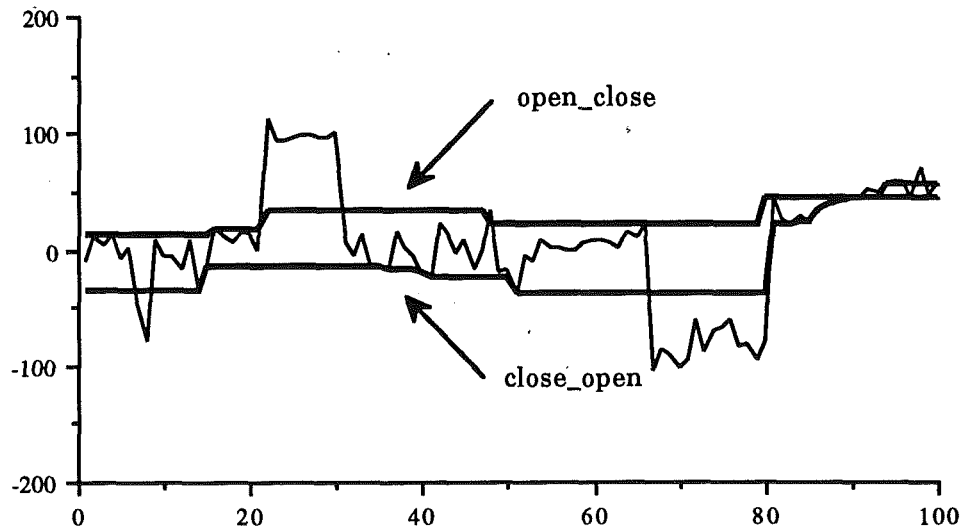


**Figure V.7.- Examples of close_open and open_close**

It is worth noticing that, since they are neither extensive nor anti-extensive, none of both filters generates an envelope of the original signal. However, close_open (open_close) filters lead to a sort of inferior (superior) envelope of the small peaks, while removing large negative (positive) peaks.

The direct detection of interior regions from the result of an close_open or open_close filter is not possible. Note that these filters remove both the texture and interior region information. However, these filters supply with a very useful tool as basis for residue transforms.

### V.2.5.- Residues with close_open and open_close

An extension of the Top Hat transform is defined by using close_open and open_close filters as basis for computing the residue [81]. These residues are not obtained directly by subtracting the original image and the result of the filtering, but a second operator is introduced in the definition. In this way, two different contrast extractors are given by:

$$\begin{aligned} &\text{Positive contrast extractor:} && I - \text{Min}(\gamma\,\varphi, I) \\ &\text{Negative contrast extractor:} && \text{Max}(\varphi\,\gamma, I) - I \end{aligned} \qquad \text{(V.8)}$$

where I stands for the identity operator. Note that the operator Min ($\gamma\,\varphi$, I) (Max ($\varphi\,\gamma$, I)) is increasing, anti-extensive (extensive) and idempotent. Thus, it is an algebraic open (close). On its turn, contrast extractors are idempotent transforms. An example of these operators is plotted in Figure V.8 where an offset (300) has been subtracted from the positive contrast extractor in order to show it jointly with the original signal and the Min ($\gamma\,\varphi$, I) result.

Residues obtained by using algebraic open (close) based on Min (Max) operators and open_close (close_open) filters nearly remove all information not dealing with peaks in the original signal. In addition, the remaining information coping with rapid fluctuations of the original signal extends in small zones and presents values smaller than in the original signal.

The extraction of both positive and negative contrasted areas can be performed by combining both residue informations. This combination turns out to be the absolute value of the residue of the identity operator and the morphological centre of the close_open, the open_close and the identity operator itself. Thus, the contrast extractor can be defined as:

$$\text{Contrast extractor: } |\, I - \text{centre}\,(\gamma\varphi,\ \varphi\gamma,\ I)\,| \,, \qquad \text{(V.9)}$$

where the centre operator stands for

$$\text{centre}\,(\gamma\varphi,\ \varphi\gamma,\ I) = \text{Min}\,(\gamma\varphi,\ \text{Max}\,(\varphi\gamma,\ I)) \qquad \text{(V.10)}$$

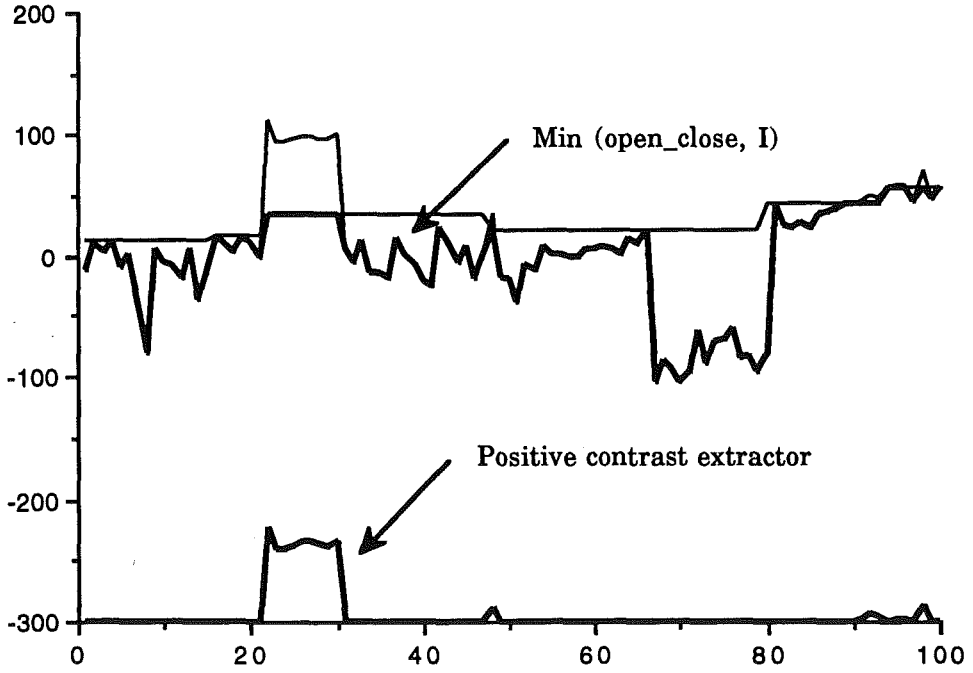


**Figure V.8.- Example of positive contrast extractor**

The centre operator is a self-dual transform. Its residue deals therefore in a symmetric way with both positive and negative peaks. Hence, the necessity of the absolute value in the definition of the residue. Figure V.9 illustrates the properties of the transforms defined above. As in the previous examples, an offset (300) has been subtracted from the residue signal in order to show all the results within the same graphic.

Owing to the self-dual property of the centre transform, its residue with the identity detects positive and negative contrasted peaks. Some rapid fluctuations of the original signal may also appear in the resulting signal. Nevertheless, as in the previous case, the range of fluctuation values is reduced by the transform. Furthermore, fluctuations appear very localised and extend through small areas comparing with peaks. The extraction of seeds marking interior regions can therefore be performed easily by means of this transform.

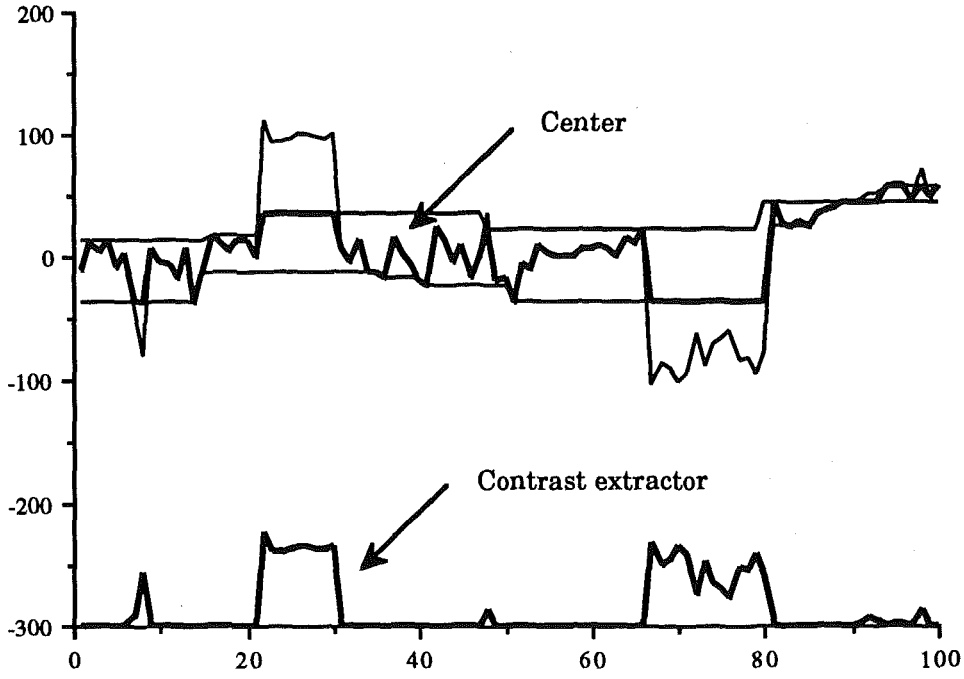


**Figure V.9.- Example of centre and contrast extractor**

## V.3.- Use of the seed image

Some points have to be considered when applying the morphological contrast extractor to the error image for creating the seed image. Seeds indicate the locations where the segmentation algorithm is allowed to create new regions. Recall that seeds should mark non-detected interior regions while avoiding textured areas. In addition, each interior region should have related as few seeds as possible. That is, in the optimum case, the seed image should present only one connected component (or seed) for each non-detected interior region.

### V.3.1.- On the choice of structuring element

In the previous section, the reasons for choosing a flat structuring element have been stated. In addition, two more features should be fixed in order to totally describe the structuring element: its shape and its size. Given that interior regions may present any kind of shape, no a priori information can be introduced in the choice of the structuring element. Hence, an assumption of isotropy is made. That is, square structuring elements are used.

Regarding the size of the structuring elements, it has to be noticed that only homogeneous regions smaller than the structuring element would be detected, as can be observed in Figure V.9. However, the use of large structuring elements would lead to the detection of regions actually being in the given segmentation besides interior regions. Note that transitions between actual regions may be marked in the error image (see Figure V.2). Hence, the size of the structuring element should guarantee the detection of all interior regions while being as small as possible.

In order to fix the size of the structuring element, the way of producing non-detected interior regions has to be studied. A non-detected interior region is a region that, being completely embedded in other region, is so small that it has been overlooked in the segmentations performed at previous levels of the multiple resolution decomposition. A square, homogeneous region of size 16x16 pixels or larger, present at the bottom of the decomposition, will, very likely, be present at the top level (l = 4). Note that, due to the decimation procedure, a pixel at the top of the pyramid is related to a square block of 16x16 pixels at the bottom image. Given that the top level segmentation algorithm is driven by an initial segmentation which assumes that each pixel is a region, such regions will be detected at this point of the procedure. Therefore, interior regions should be smaller than this size, in order not to be detected.

However, as it has been emphasised in Chapter IV, final segmentations performed at the top of the pyramid are close to initial ones; that is, only a few refinements are carried out. Thus, initial segmentations at level (l = 3) are close to be totally formed by square regions of size 2x2. Note that, with square regions of this size, no interior region can be defined. That is, a region cannot be in the interior of a 2x2 pixels region but only lay on its contour. Given that, level (l = 3) is taken as the reference level and the size of the largest possible non-detected interior region at the bottom level is 8x8 pixels. Actually, structuring elements of size 7x7 pixels are large enough to detect interior regions at the bottom level. In an analogous way, 5x5 structuring elements are used at level (l = 1) and 3x3 ones at level (l = 2). Following a conservative policy, contrast extraction is also carried out at level (l = 3) with a 3x3 structuring element. Note that this step implies the filtering of a 32x32 pixels image, which requires very little computational load.

Figure V.10 shows the contrast extraction performed with a square, 7x7, flat, structuring element on two different images. These images are the error image of Figure V.2 for the Cameraman image, and the error image obtained with the segmentation of Miss America image shown in Figure IV.19. The

gray level values have been inverted in such a way that zero values after contrast detection are represented by a gray level value of 255.

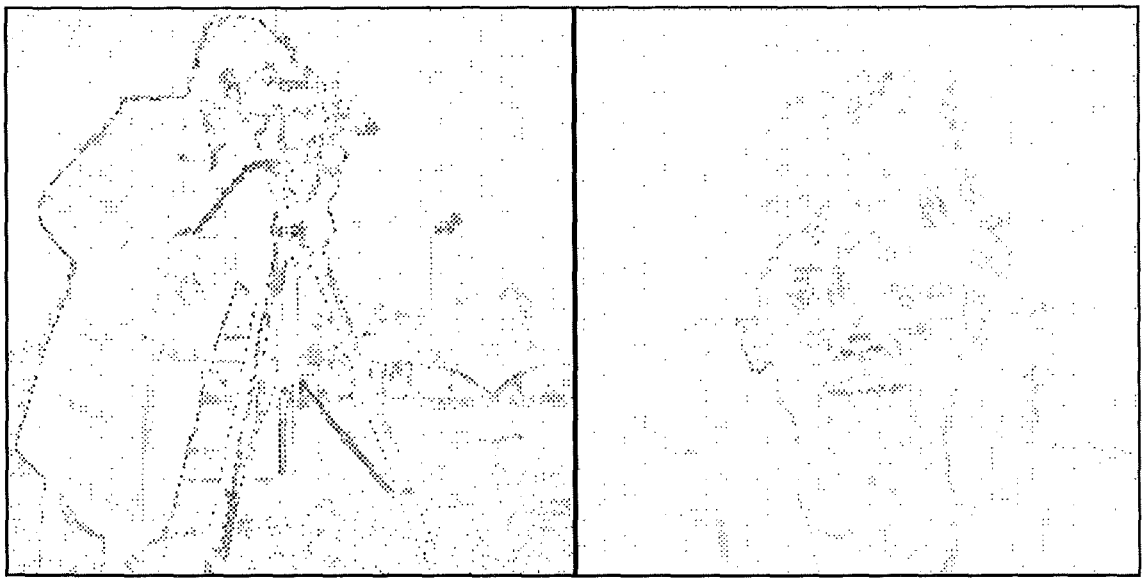

**Fig. V.10.- Examples of contrast extraction**

Note that all the details of the error image are presented after the contrast extraction. Moreover, information regarding textured areas (e. g.: the grass in the Cameraman image or the hair in the Miss America image) is mainly represented by a few isolated elements of small value. Therefore, the extraction of non-detected interior regions can be easily performed from this result.

### V.3.2.- Cleaning the seed image

Three different cleaning steps have to be performed on the contrast extractor result in order to obtain the seeds. The seed usefulness is to mark the positions within the image where the segmentation algorithm may create new regions but it has not been allowed to. Thus, seeds should not be related to pixels laying on the region boundaries of the given segmentation, since these positions are already checked by the segmentation algorithm. Given that, a first step removing all components laying on the region boundaries is carried out.

The second cleaning step relies on the gray level information of the elements detected by the contrast extractor. As shown in the previous section, low valued elements correspond to textured areas whereas elements with high

values are more likely related to interior regions (see Figure V.9). Hence, a threshold is applied so that small valued texture elements are removed. Both dynamic and global thresholding have been studied. When using dynamic thresholding, the information of the estimated variance of each region within the given segmentation has been used to adapt the threshold locally to the characteristics of the error image. Note that the variance of a region in the segmentation and in the error image are the same.

Results obtained with both techniques are very alike, global thresholding requiring much less computations than dynamic one. Furthermore, final results are very robust to the choice of global thresholds. However, a conservative threshold has to be chosen in order not to split information dealing with a single interior region into several seeds. A threshold value of 20 has been used in this work. In this way, the creation of more than one seed for each region is prevented.

The last cleaning step is based on the size of the detected elements. Small elements in the contrast extraction are more likely related to textured areas rather than to actual interior regions (see Figure V.9 and Figure V.10). Hence, the removal of small elements. However, final elements will act in the segmentation as possible seeds for new regions. That is, the segmentation algorithm can expand or shrink the given seed, relying on whether the new region leads to a realisation of the hierarchical model increasing the joint likelihood or not. Thus, and following the above conservative policy, only very small elements are removed from the seed image, leaving the final decision to the basic segmentation algorithm itself.

Components having one or two connected pixels are therefore removed. The removal is carried out by using a directional open filter with a line segment of size 3 as structuring element. Four different opens are performed in this filter, using for each one a different orientation of the structuring element (0º, 45º, 90º and 135º, respectively). The final result is obtained by taking the superior of the four previous results.

The whole procedure for obtaining the set of seeds is illustrated in the block diagram of Figure V.11. Note that all steps in the procedure have been grouped in a single block. In the sequel, the whole procedure will be represented by this block.

Examples of final seed images obtained after applying the whole cleaning procedure are shown in Figure V.12. Regarding the Cameraman

image, it is worth noticing that very few seeds have been found in the grass. That is, the algorithm for extracting seeds performs a correct discrimination between interior region and texture information. Moreover all relevant details in the error image are correctly represented in the seed image. Every interior regions have related a unique, connected seeds.

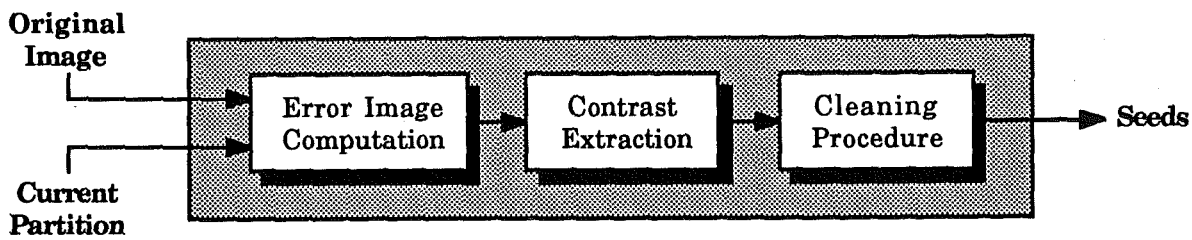


**Fig. V.11.- Block diagram of the seed extraction**

An analogous analysis can be made with respect to the Miss America seed image. It should be highlighted that no seeds appears in the zones of the background or in the pullover. All seeds are devoted to mark lacks of detail in the face and the hair.

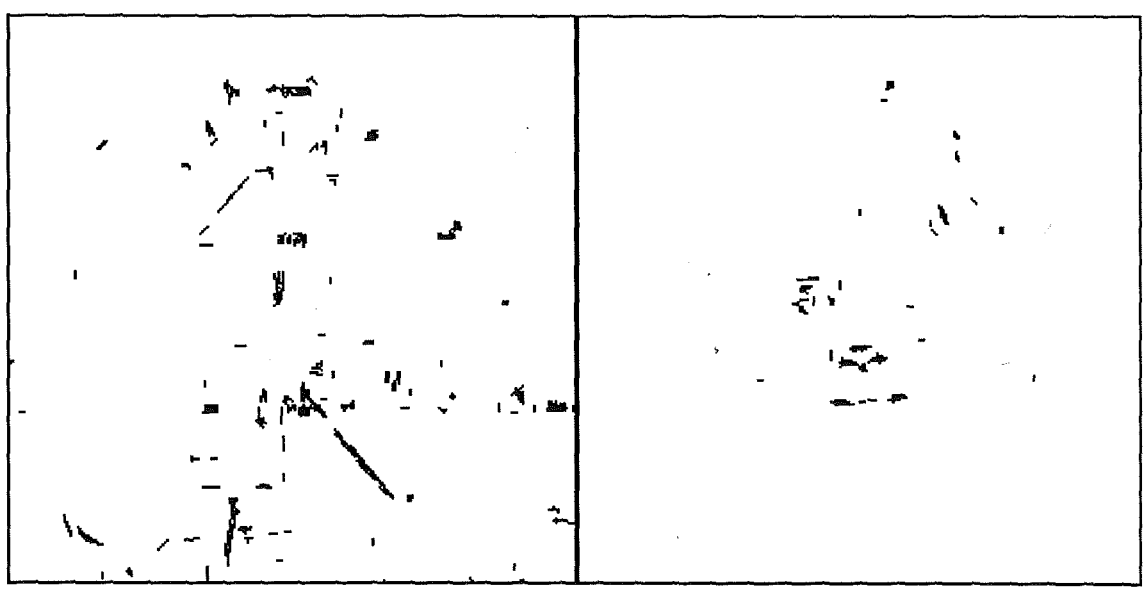

**Fig. V.12.- Examples of final seed image**

### V.3.3.- Using the seeds in the multiresolution segmentation procedure

Two different techniques have been studied for introducing the seed information within the segmentation procedure. In both cases, seeds are introduced at each decomposition level in order to refine the previous segmentation. That is, the error image at level (l) is computed by the difference between the original image at this level (l) and the segmentation supplied by the final algorithm presented in Chapter IV.

The first technique uses the seeds as a set of marks to guide the segmentation algorithm to the positions where new regions can be created. That is, the segmentation method proposed in Section V.1 is restricted to perform label changes in pixels either laying on the boundary of regions or contained in the seed image. Although the quality of the segmentation results improves, this technique arises some problems, as pointed out in Section V.1.

New regions are created starting from a single pixel. As it has been emphasised through the different chapters, parameter estimations performed on few samples are not reliable. In this case, the situation worsens since the variance estimation of Chapter III yields a zero value when it is carried out on only one sample. Therefore, to determine whether a new region has to be created or not relying on such estimations may lead to erroneous decisions.

Another drawback of this technique is that, in spite of having each interior region detected as a single, connected seed, these regions can be eventually oversegmented. That is, each single pixel in a seed can generate a new region. As pointed out in Section V.1, this multiple seed effect depends on the shape of the interior region and on the scanning procedure.

On the other hand, the procedure for creating new regions relying on the seed information has to be emphasised. The few possible seeds related to textured areas are overlooked by the algorithm. That is, either the algorithm does not create any region at these locations or, if it creates some, they are not expanded. Therefore, they remain small and are eventually removed by the segmentation algorithm. On the contrary, an interior region may be oversegmented but the pixels in its seed are expanded up to covering, at least, the whole initial seed.

The above observations lead to a second technique for introducing the seed information in the segmentation procedure. Given that the algorithm expands the chosenpixels in the seeds up to covering the whole seed, seeds

could be introduced directly as new regions in the previous segmentation. In this way, the segmentation algorithm is allowed to decide whether this new region should be expanded, shrunk or removed.

This technique leads to carrying out twice the same segmentation algorithm at each level. In the second stage, the algorithm is driven by the an initial partition resulting from adding the information obtained from the previous segmentation and the information obtained from the seed image. However, very few label changes are performed in this second stage, mainly related to refinements on new region contours. That is, the segmentation procedure mainly looks for the actual shape of regions coming from the seed information. Typically, this segmentation stage only needs one scanning in order to finish the whole procedure.

A study has been performed on testing if the direct introduction of new regions in the segmentation leads to partitions with higher likelihood. It has been seen that almost all seeds, when added to the previous segmentation, lead to a realisation of the model with higher probability than that of the previous segmentation. Moreover, in the cases where a new region leads to a lower probability realisation, the algorithm removes this region and recovers the initial state.

The second technique for introducing the seed information in the segmentation algorithm has been therefore assumed. In this way, the proposed final multiresolution segmentation scheme is illustrated as a block diagram in Figure V.13. In this figure, the notation followed in Chapter IV is used. In addition, the set of seed images has been denoted by MP(l).

Note that the block devoted by Monolevel Segmentation is driven, in a first stage (a), by the segmentation provided by the previous level and, in a second stage (b), by the information generated by the Seed Extraction block and the Monolevel Segmentation block itself. Therefore, the Seed Extraction and the Monolevel Segmentation blocks do not create a recursive loop, but a path which is followed only once. The introduction of the Seed Extraction block in the procedure translates into increasing the computational load of the whole method in less than 10%. Segmentations are therefore obtained in a feasible amount of time.

The segmentation scheme of Figure V.13 is further illustrated in Figure V.14, where the different steps of the segmentation of the Cameraman image are shown. In the first column, the set of seed images -MP(l)- is shown as well

as the mosaic images of the segmentation of the four highest levels of the pyramid. In the second column, the contour images of the whole decomposition are shown as well as the four highest levels of the original pyramid.

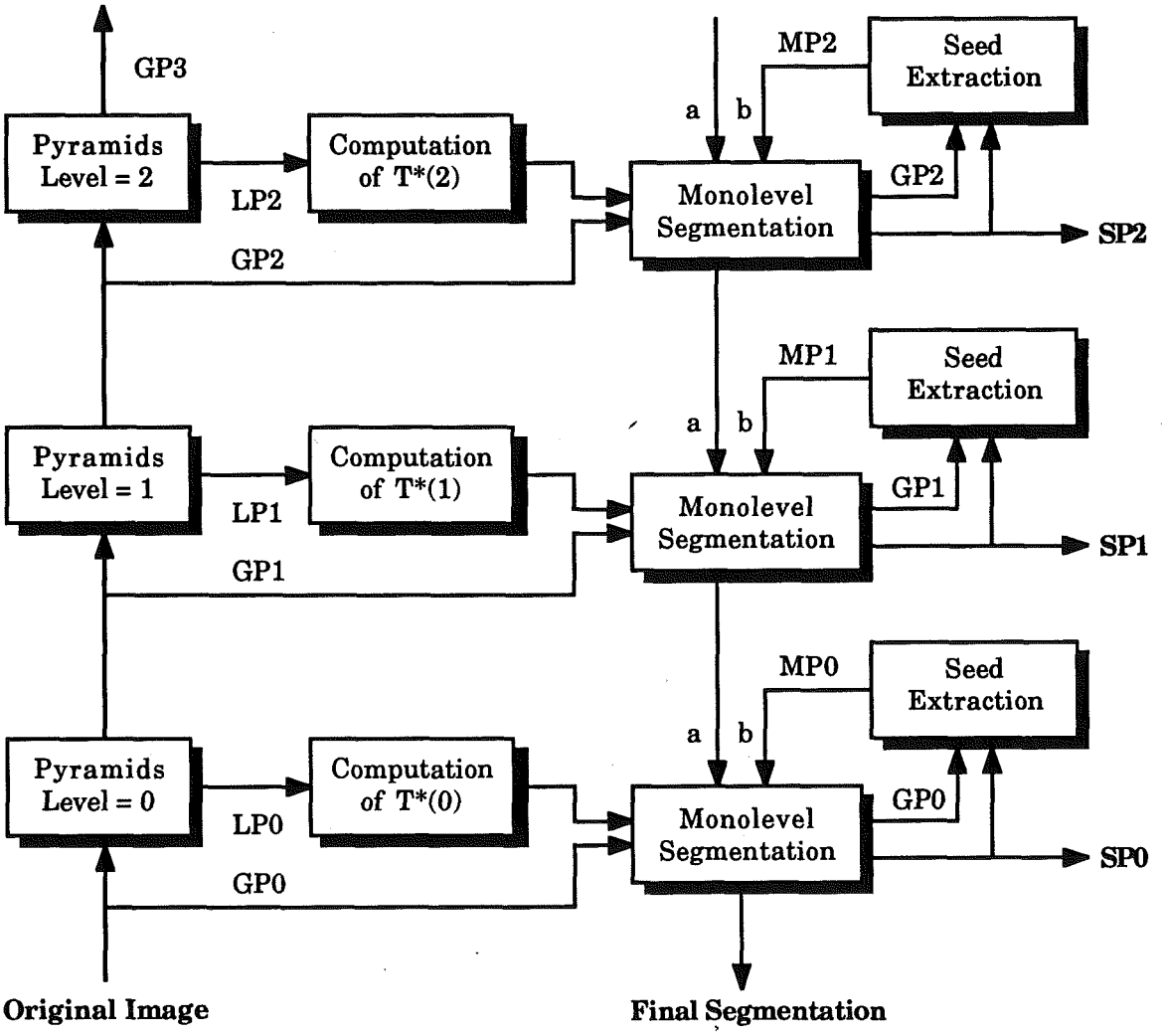


**Fig. V.13.- Final multiresolution segmentation scheme**

High level seed images contain zero (l = 4, l = 3) or very few seeds (only three in l = 2). Moreover, almost no components are detected in textured areas. Note that there are only four texture components in the lowest level seed image (two at the left side of the tripod left leg and two at the bottom, right side corner) and zero in the rest of seed images. Furthermore, from this set of seeds, only one of them is expanded and forms a region on the final segmentation. That is, the other three components are shrunk and eventually removed from the final segmentation.

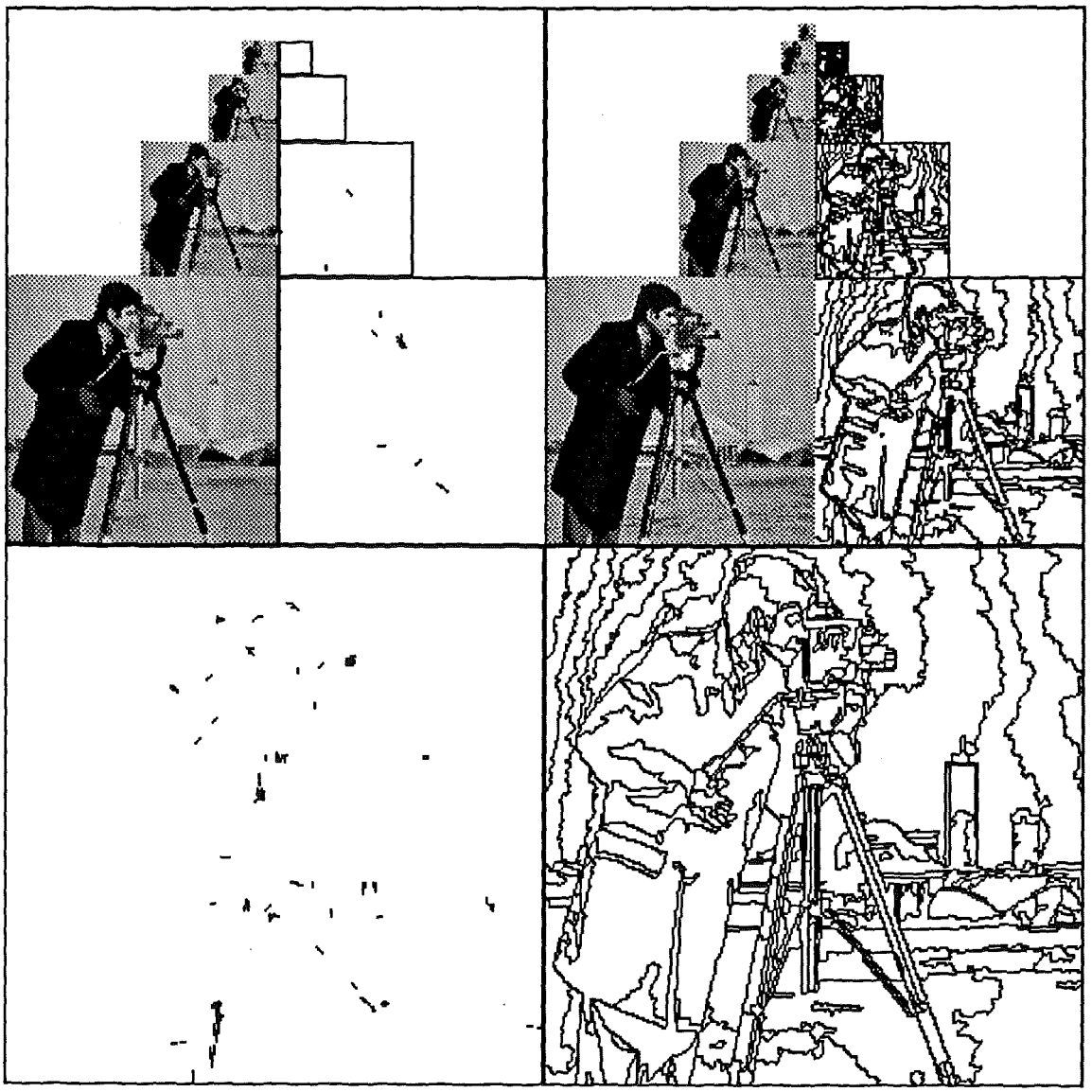

**Fig. V.14.- Example of seed image and segmentation**

## V.4.- Segmentation results

This section is devoted to the presentation of final segmentation results achieved on a large set of images. Each example is illustrated by means of a figure containing the whole Gaussian pyramid of the original image as well as the segmentation pyramid presented as a set of contour images. Finally, the mosaic image of segmentation at the bottom of the decomposition is shown. For each case, the computational load of the whole procedure is given. This load is quantised in terms of CPU time in a Sun Sparc II workstation.

### V.4.1.- Cameraman image

The final segmentation presented in Figure V.15 contains 353 regions and has required a computational load of 30 seconds. It is worth noticing that almost all relevant details in the image are preserved in the segmentation, while gathering the textured areas in a few regions. As commented in previous chapters, the correct segmentation of textured areas is owing to the use of a multiple resolution approach. Nearly all the regions forming the textured areas are already present in the segmentation of the highest levels of the decomposition. Moreover, note the high quality in the reconstruction of the different parts of the camera where no main feature has been overlooked.

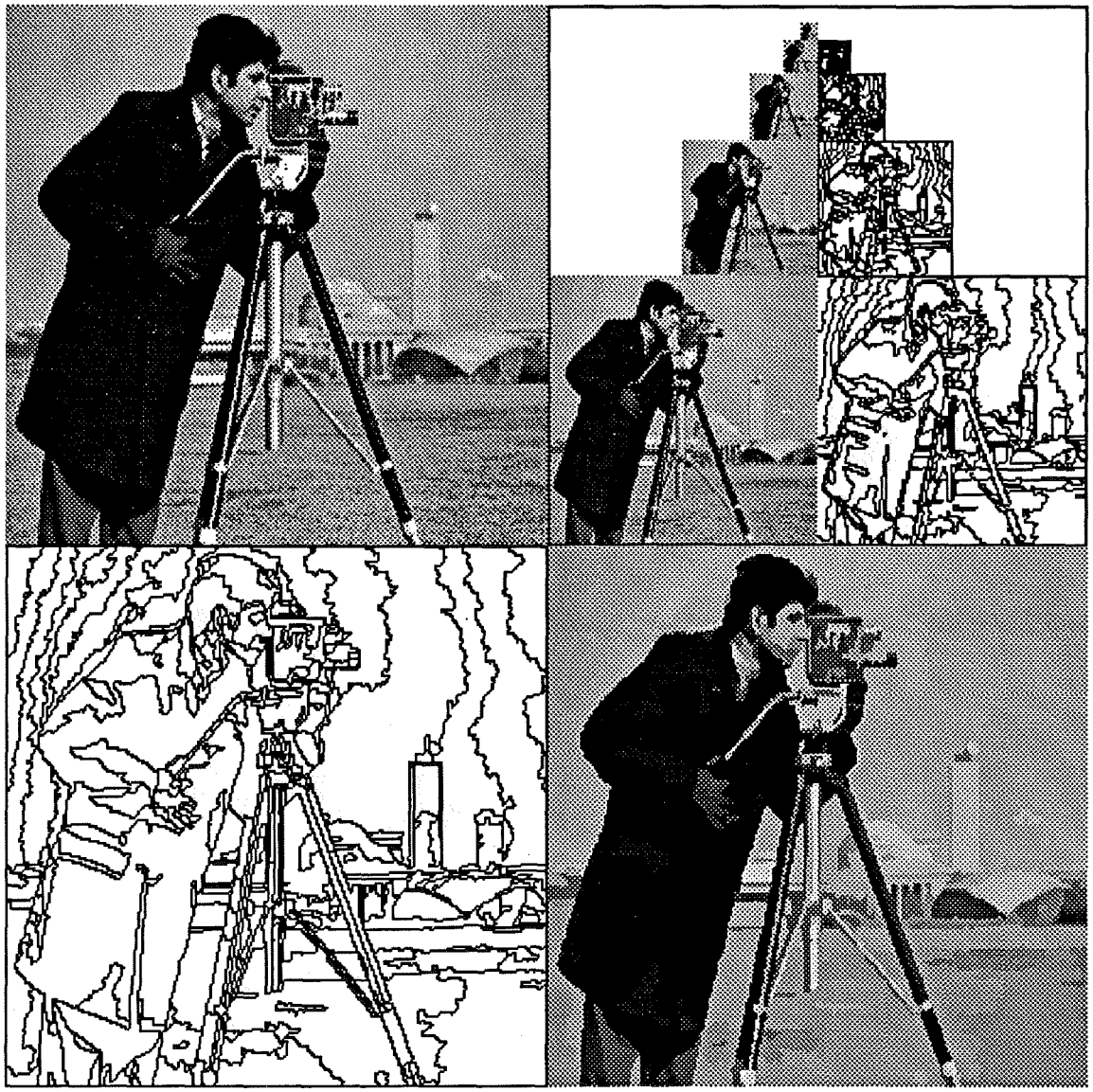

**Fig. V.15.- Final segmentation of the Cameraman image**

### V.4.2.- Miss America image

The final segmentation presented in Figure V.16 contains 180 regions and has required a computational load of 22 seconds. The quality of this final segmentation has also improved with respect to that of Figure IV.19. Details which are not detected in the segmentation of Figure IV.19 have been separately segmented in this image (e. g.: the the teeth and the left eye). Moreover, the bright zones on the right side of the hair are correctly reproduced. In addition, the hair and the background are still segmented into different regions.

**Fig. V.16.- Final segmentation of the Miss America image**

### V.4.3.- Cars image

The final segmentation presented in Figure V.17 contains 505 regions and has required a computational load of 36 seconds. Note that nearly all the details in the original image are represented in the final segmentation. In this case, even the blurred numbers and the central black dot of the wheels have been detected. A fair part of regions in the final segmentation is devoted to the blurred background. The blurring introduces a huge amount of uncertainty in the image and the segmentation needs a large number of regions for coping with it.

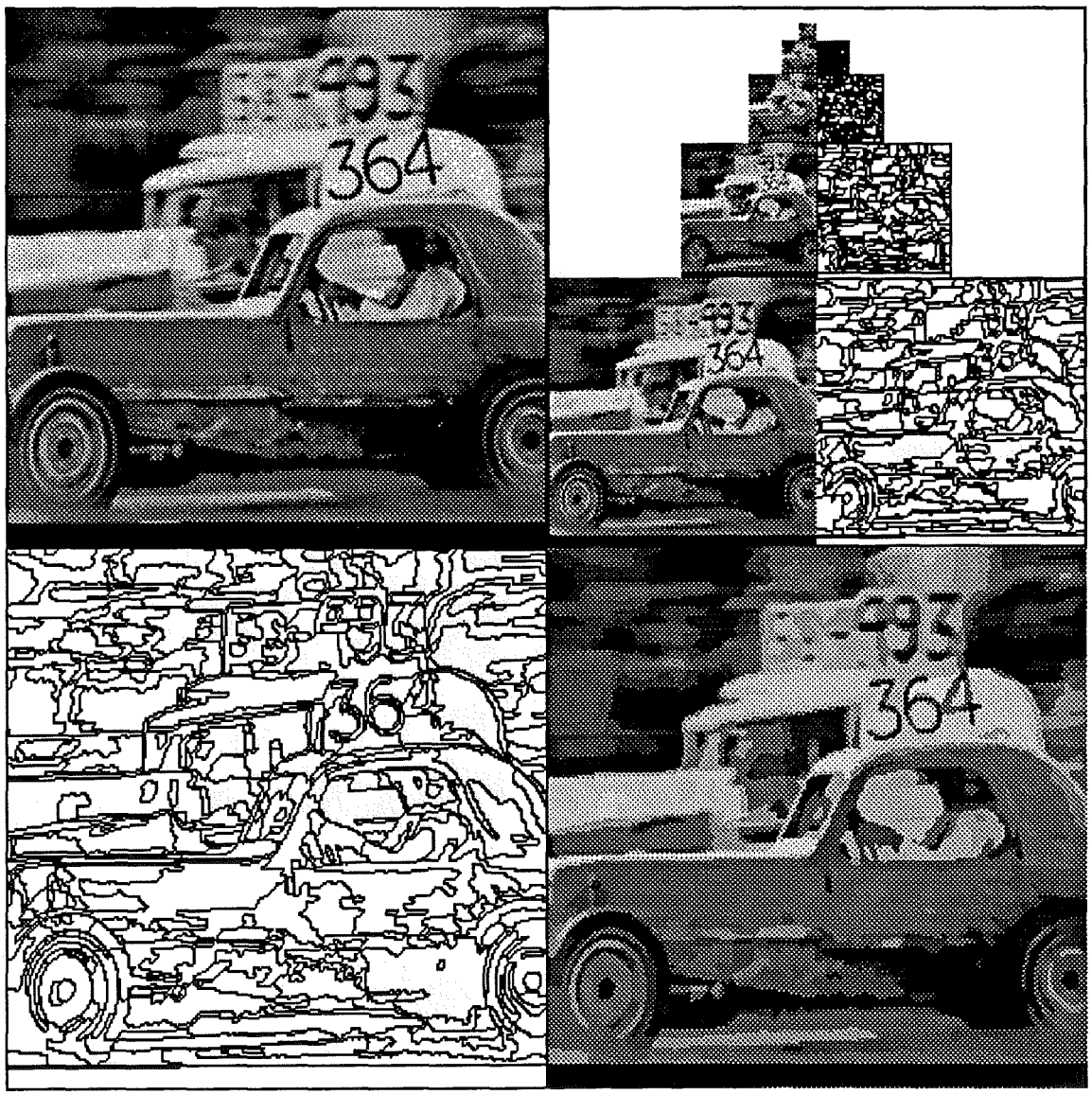

**Fig. V.17.- Final segmentation of the Cars image**

### V.4.4.- Building image

The final segmentation presented in Figure V.18 contains 429 regions and has required a computational load of 33 seconds. It has to be emphasised that all the windows of the building are correctly represented in this segmentation. Note that all details in the original image are detected, besides some parts of the horizontal, dark lines appearing in the column and the wall. These lines have not been detected mainly given that they are not completely connected. Therefore, the algorithm detects several small, unconnected regions which are finally removed.

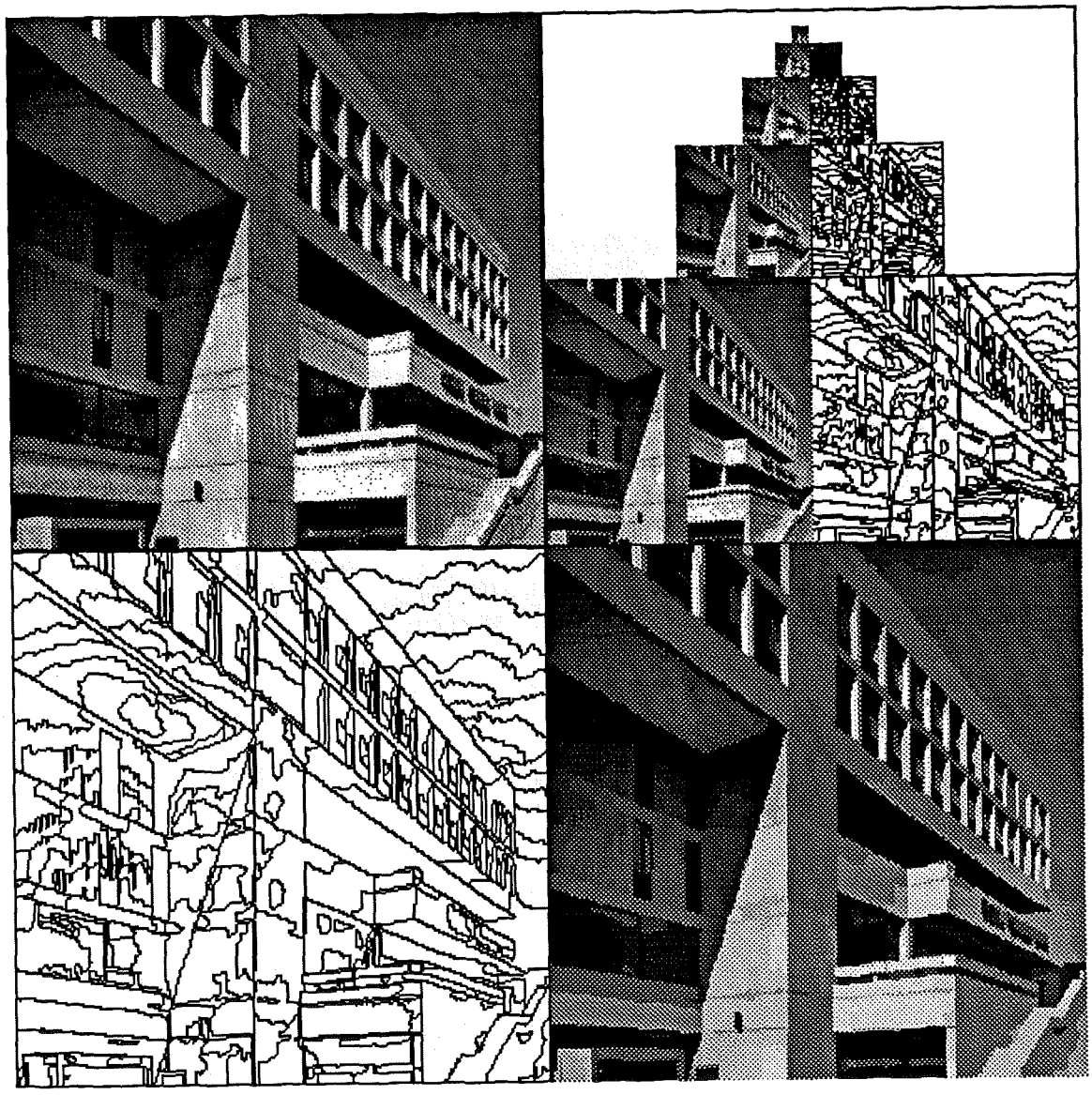

**Fig. V.18.- Final segmentation of the Building image**

### V.4.5.- Peppers image

The final segmentation presented in Figure V.19 contains 374 regions and has required a computational load of 32 seconds. The main problem when segmenting this image is the possibility of overgrowing regions due to the contact between the different elements in the image. Note that this effect seldom happens and almost every region is contained in a single fruit. Moreover, in spite of using a texture representation as basic as a constant gray level per region, the mosaic image correctly represents the volumes of fruits in the image. However, note that such a simple representation fails in characterising some shadowed areas.

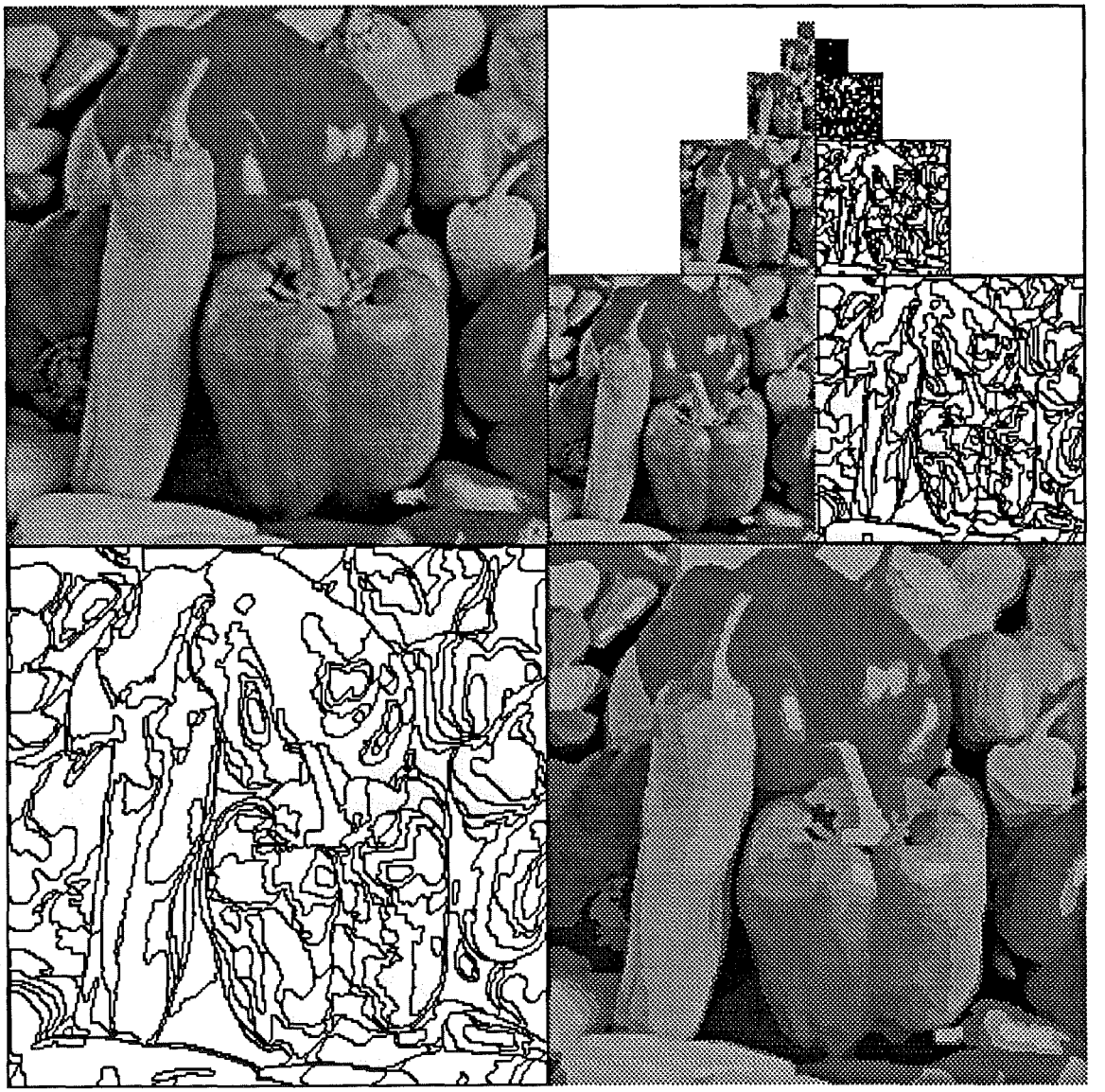

**Fig. V.19.- Final segmentation of the Peppers image**

### V.4.6.- Lena image

The final segmentation presented in Figure V.20 contains 270 regions and has required a computational load of 20 seconds. This is a rather complicated image given that it contains different textures, several details, homogeneous zones and quite blurred areas. It should be noticed that the most textured area (the tail of the hat) has been split only into few regions. In addition, details in the face (as the eyes or the nose) have been correctly detected. Furthermore, only a small part of the hat has been connected with the background, in spite of the gray level similarity. As in the Cars image, blurred areas yield several regions.

**Fig. V.20.- Final segmentation of the Lena image**

### V.4.7.- Bridge image

The final segmentation presented in Figure V.21 contains 328 regions and has required a computational load of 39 seconds. As in the previous case, this is a complex image. It contains very fine structures that can be understood as textures and, due to the fog, all the image is rather blurred. Note that the fine structures have been gathered in a few regions. That is, the use of the seed information has not yield small regions in the area which would have spoiled the global segmentation. Moreover, small, relevant details on the bridge have been correctly segmented (e. g.: the cable or the white dot between the bases of both towers).

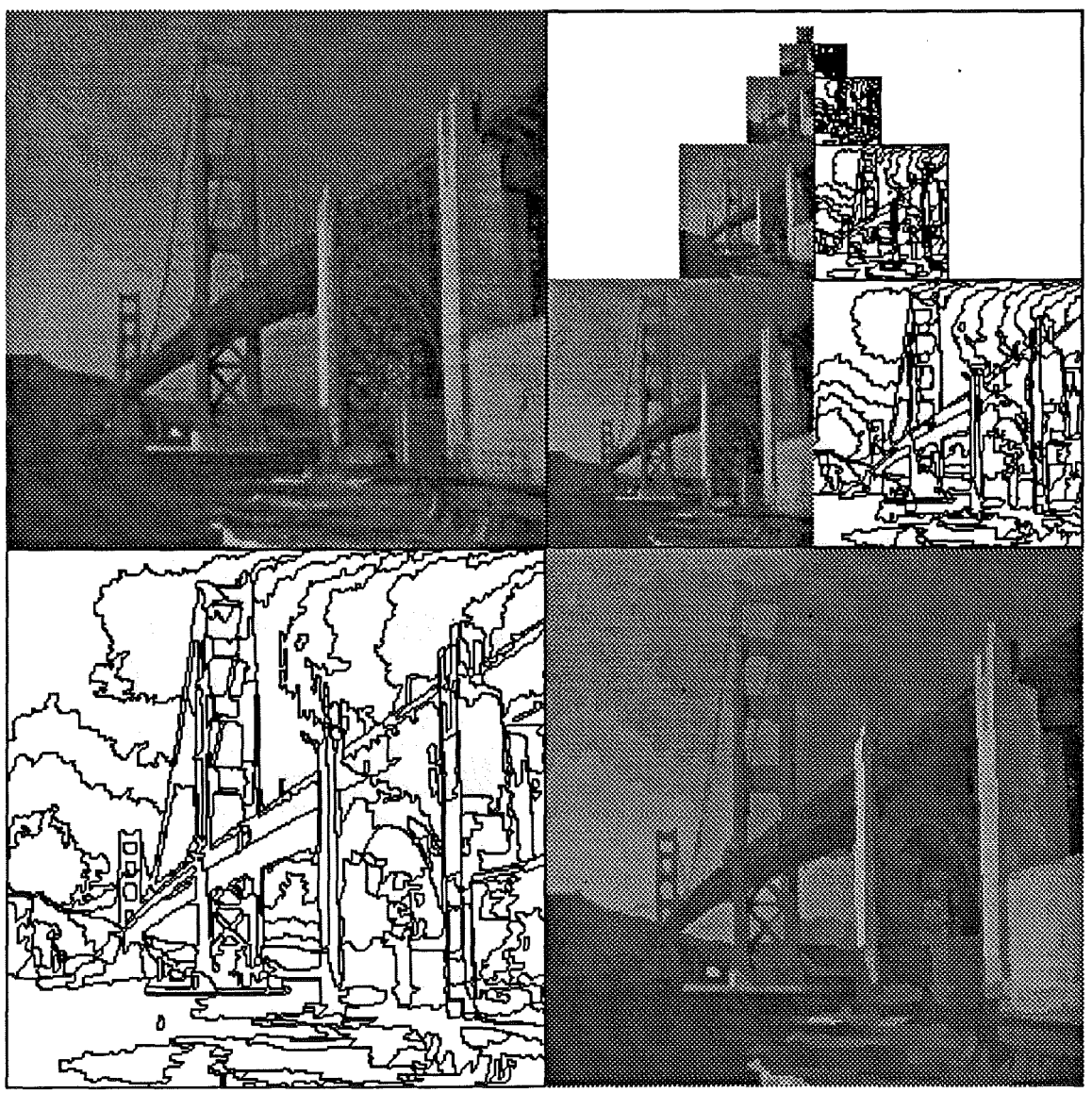

**Fig. V.21.- Final segmentation of the Bridge image**

### V.4.8.- Aerial image

The final segmentation presented in Figure V.22 contains 328 regions and has required a computational load of 18 seconds. In this case, the final segmentation is very close to that performed by means of the algorithm presented in Chapter IV. The use of the seed information has introduced only a few regions in the forest area associated to shadows of the trees. Furthermore, the segmentation of the crops is very alike either using seeds or not. That is, the texture presented in the crop areas has been overlooked by the seed extraction.

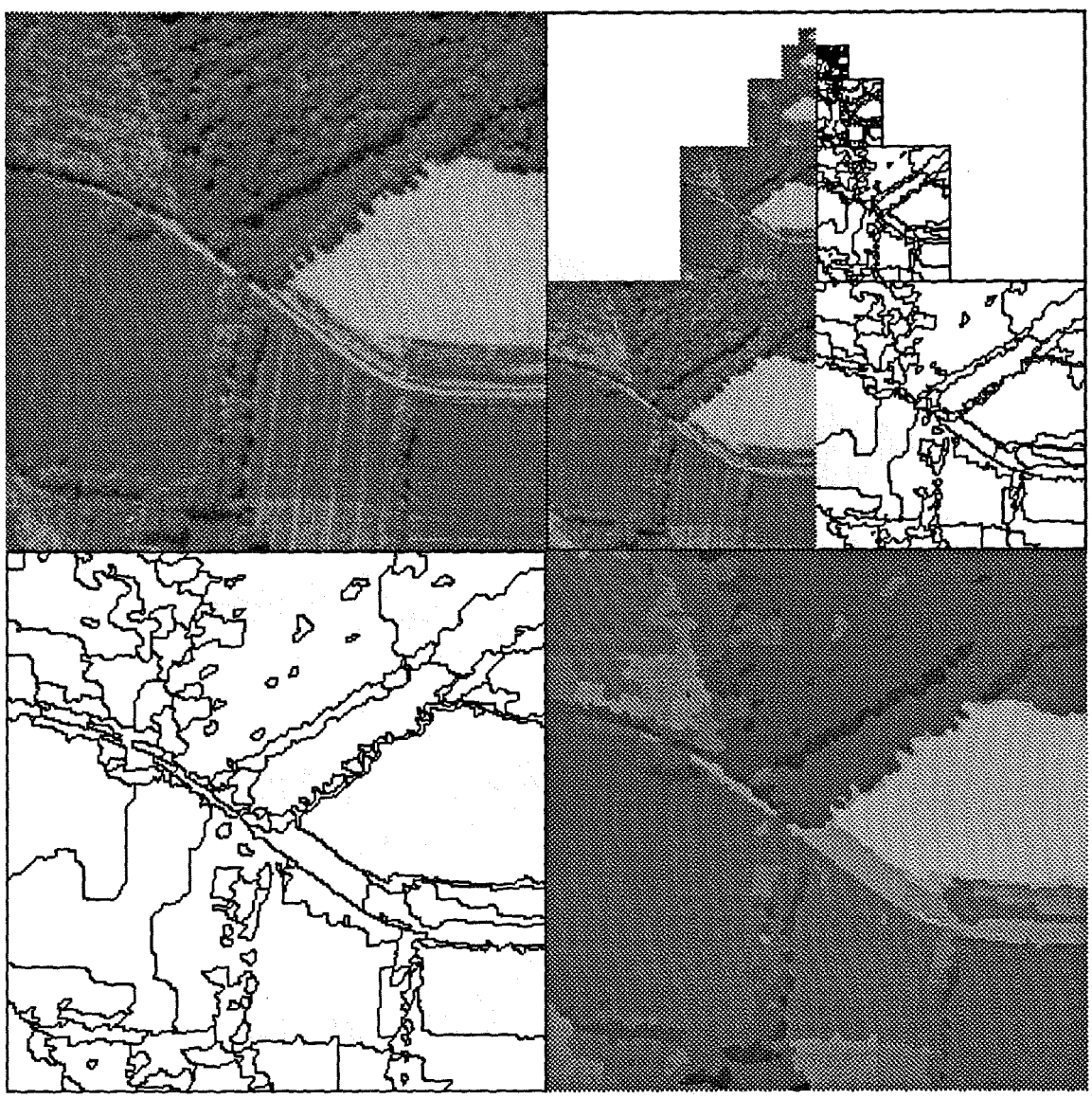

**Fig. V.22.- Final segmentation of the Aerial image**

### V.4.9.- Synthetic image

The final segmentation presented in Figure V.23 contains 33 regions and has required a computational load of 19 seconds. As in the Aerial image case, the use of seed information does not change the behaviour of the segmentation algorithm when handling textured areas. The final number of regions is almost the same in both cases (the segmentation in Chapter IV contains 34 regions).

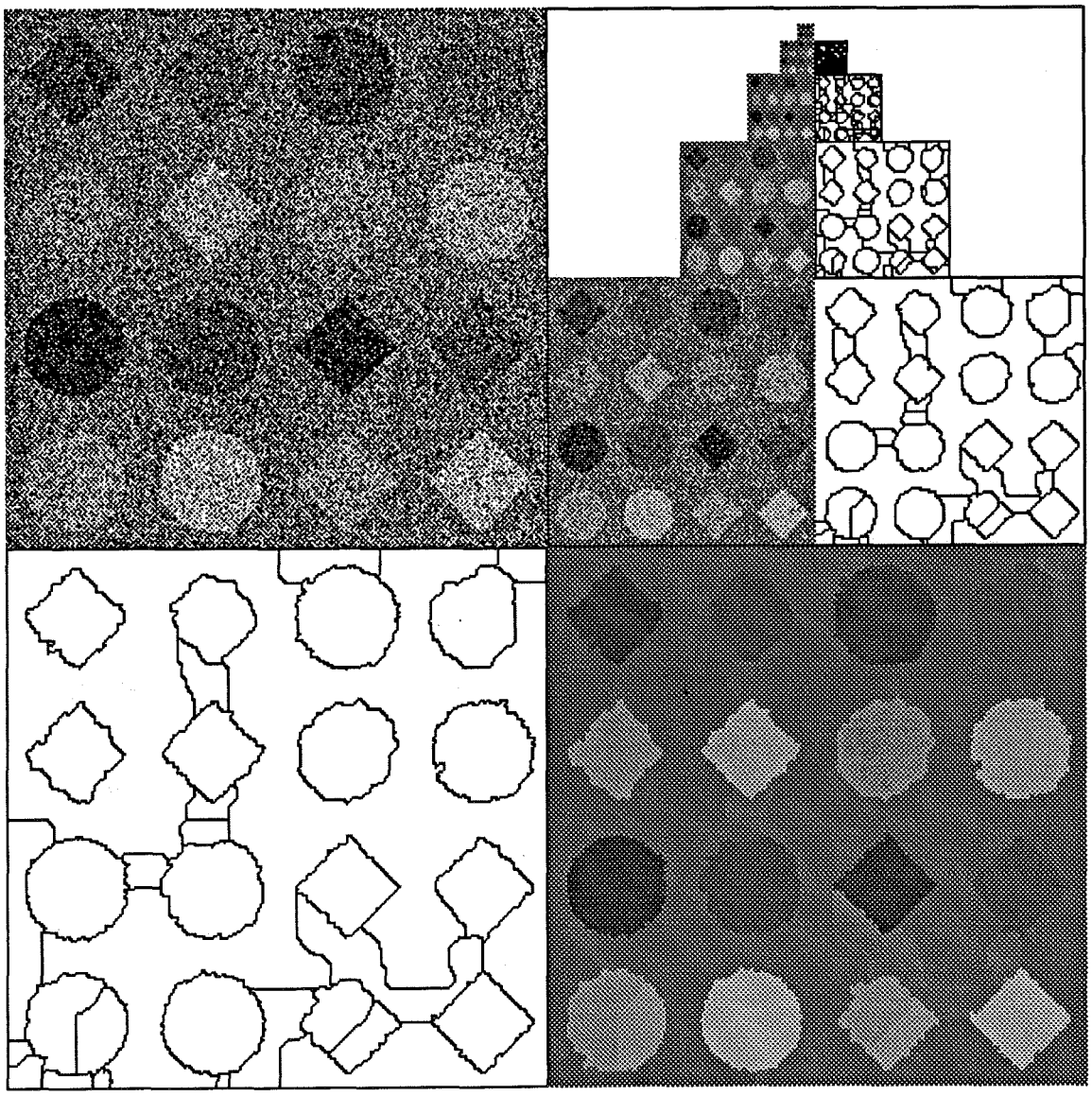

**Fig. V.23.- Final segmentation of the Synthetic image**

### V.4.10.- Table-Tennis image

The final segmentation presented in Figure V.24 contains 75 regions and has required a computational load of 17 seconds. Note that the use of seeds has neither spoiled the segmentation of textured areas in this case. That is, no new region has appeared in the background. On the contrary, the segmentation of the hand and the racquet presents more details in this case. Dark regions due to the shadows of the fingers, as well as a white spot in the handle of the racquet have appeared.

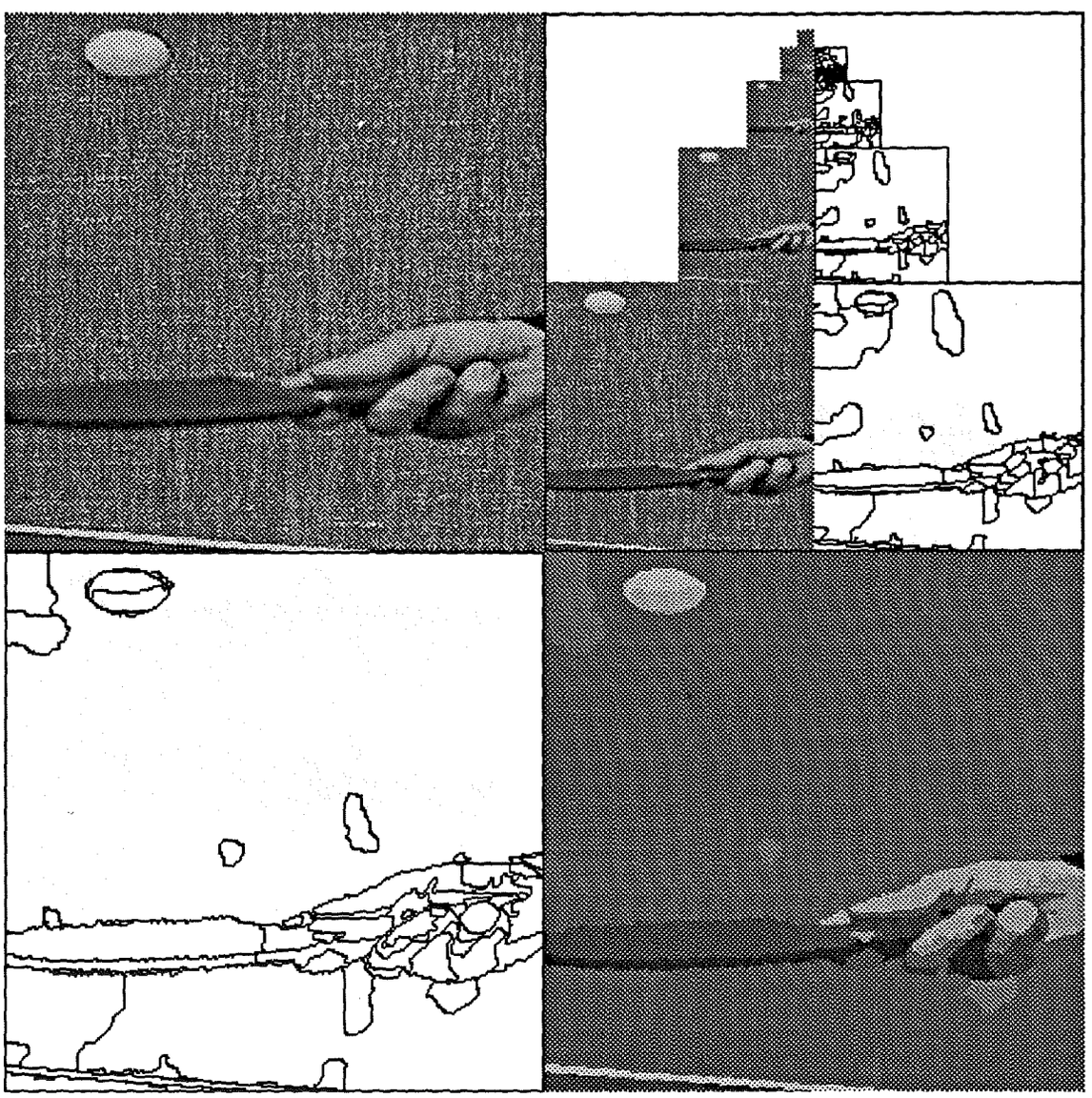

**Fig. V.24.- Final segmentation of the Table-Tennis image**

### V.4.11.- Seurat image

The final segmentation presented in Figure V.25 contains 495 regions and has required a computational load of 105 seconds. The high computational load as well as the presence of a new level in the decompostion is owing to the fact that the size of the original image is 512x512 pixels. Several details in the picture have been correctly segmented (e.g.: the eye of the horse or the scarf in the right side of the picture), besides the shape of the dancer. This region has been overlooked since the brush-strokes in the picture produce a soft gradient conecting the body of the dancer with the background.

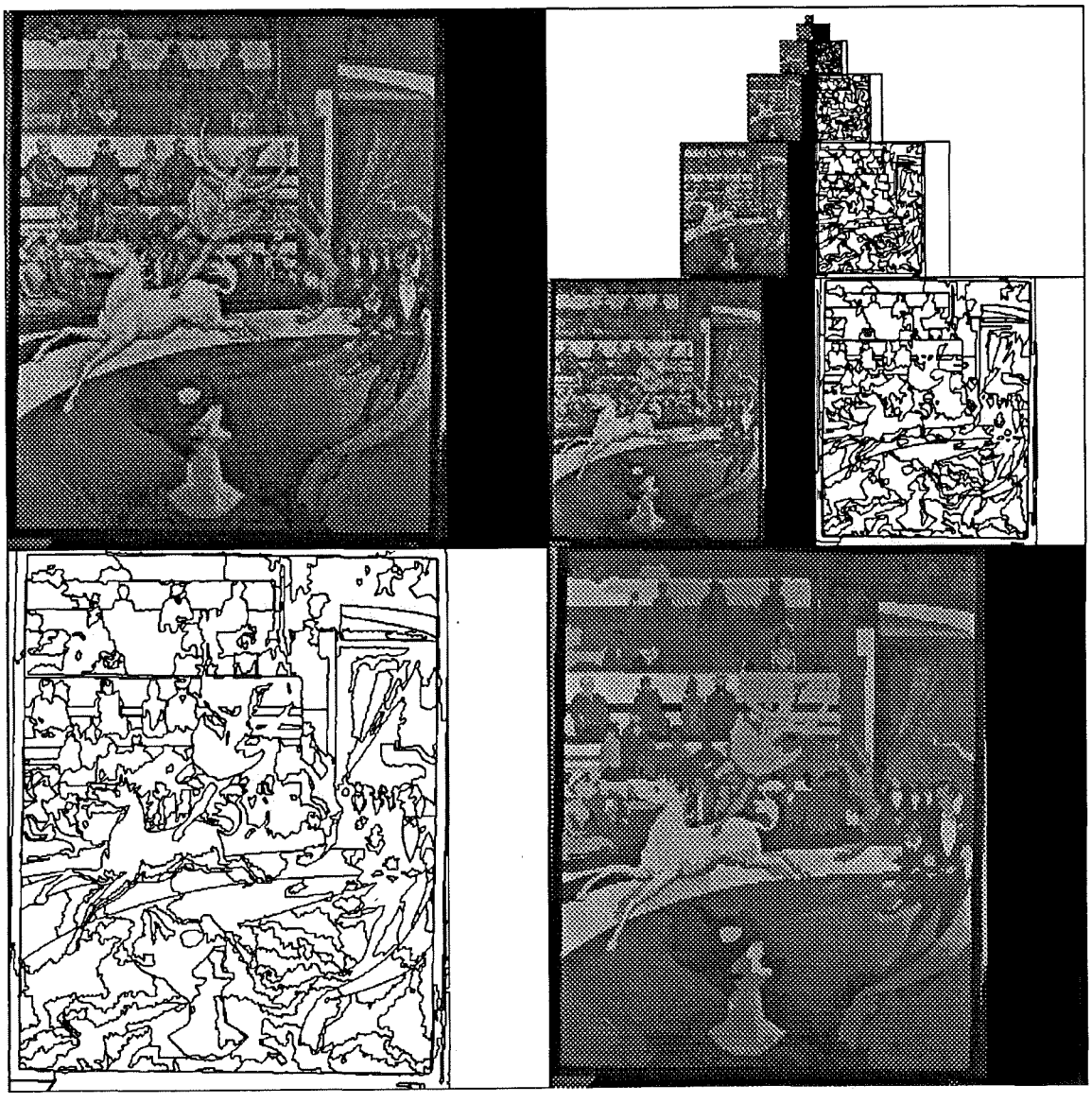

**Fig. V.25.- Final segmentation of the Seurat image**

### V.4.12.- Anevrism image

The final segmentation presented in Figure V.25 contains 756 regions and has required a computational load of 107 seconds. As in the previous example, the size of the original image is 512x512 pixels. Note that the main structures of the image have been correctly segmented. Although the veins are not detected as a single region, almost all of them are represented in the mosaic image. Furthermore, small, bright objects in the interior of the dark area have been extracted.

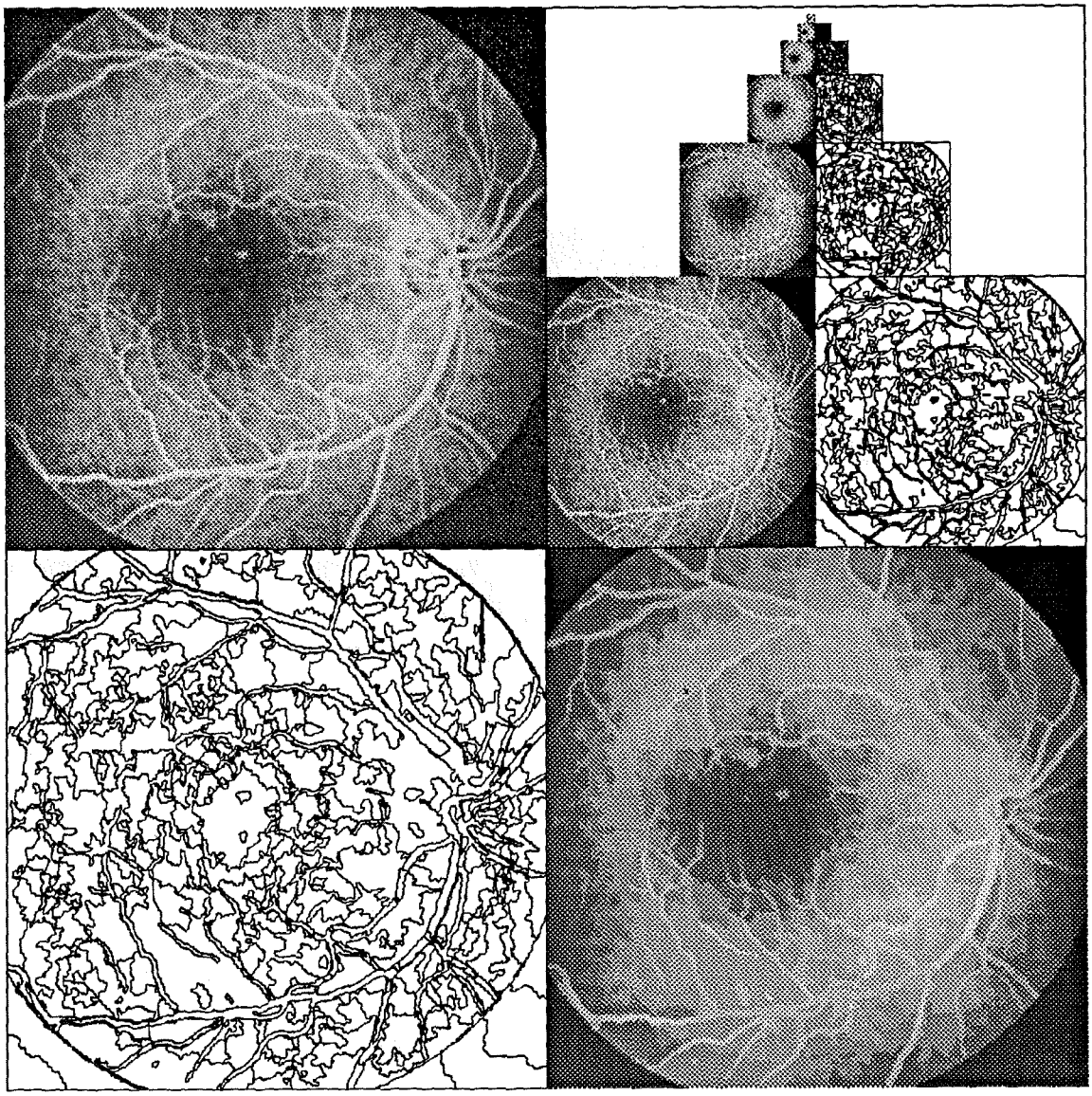

**Fig. V.26.- Final segmentation of the Anevrism image**

As it has been seen, the segmentation technique of Chapter IV has been extended to deal with non-detected interior regions. The method for extracting such regions has been shown to be efficient. Moreover, it does not spoil the segmentation of textured areas and does not need the introduction of any kind of information. Thus, the algorithm can be carried out in an unsupervised way. Furthermore, it has been tested with a large set of very different images achieving high quality results in all cases in a feasible amount of time (the average computational load is of 27 seconds for images of 256x256 pixels).

## V.5.- Summary

This chapter has been devoted to the extension of the multiresolution segmentation technique presented in Chapter IV to detecting interior regions. An interior region is a region such that its contour does not touch the contours of other regions at any point. Interior regions are not detected by the above technique given that the segmentation is mainly performed as a contour refinement, without checking pixels in the inside of regions. Therefore, in order to detect interior regions, the algorithm is allowed to create new regions at every pixel of the image.

In spite of the success of this method in detecting interior regions, it has been withdrawn since it arises too many problems. Although interior regions are detected, the algorithm oversegments them. Moreover, improvements achieved in segmenting textures by using multiresolution approaches are spoiled, since priority is given to local analyses of the image when allowing to create new regions. Hence, the necessity of providing the algorithm with information marking where the creation of new regions is allowed and where is not (concept of seed). In addition, interior regions should be related to either only one seed or as few as possible, in order not to oversegment them. All this information is gathered in a binary image called seed image.

Towards the goal of obtaining these seeds, the concept of error image is used. Error images have been defined with respect to a given segmentation. In this way, an error image is the difference between an original image and the mosaic representation of its segmentation. Therefore, assuming that regions in the segmentation are correctly located, error images contain mainly information about textures of the original image and about interior regions not detected by the segmentation. Actually, contour information does appear in error images owing to the uncertainty present in original images. However, this information can be easily removed.

Discrimination between both kinds of information could be performed by segmenting the error image. However, this approach turns out to result in cumbersome algorithms which demand, at least, twice the computational load of the previous multiresolution segmentation technique. A simpler approach can be used by taking advantage of the fact that interior regions appear in the error image as nearly flat, contrasted areas, whereas textured information appears as dense, contrasted fluctuations. Morphological tools are applied for detecting such areas, given that they are known to efficiently perform in discriminating between these kinds of information.

Basic morphological tools and their main properties have been analysed. This analysis has been carried out from the point of view of their application to the problem of detecting interior region from error images. In this framework, basic morphological transforms as morphological filters do not solve the problem and, therefore, more elaborated operators have to be used. In this way, morphological residues are studied. These operators are defined as being the difference between two morphological filters.

From the set of different residues that can be defined, the difference between the original signal and its centre (contrast extraction transform) turns out to be very efficient for detecting interior regions. The centre is computed from the open_close, the close_open and the identity operators. The efficiency of this transform results from the fact that open_close filters yield signals which mainly remain below positive meaningful peaks (bright interior regions) and above rapid fluctuations (textured zones). An analogous result can be stated for close_open filters. Moreover, since a centre transform is used, the result is self-dual; that is, it deals in a symmetric way with positive and negative information.

Some points related to the application of the contrast extraction transform for obtaining the seeds have been discussed. Regarding the structuring element, flat, square structuring elements have been chosen. The reason for chosing flat structuring elements is mainly owing to their capability for preserving edges and allowing fast implementations. On its turn, the fact of being square is due to a lack of a priori information about the shape of the elements to be detected. The size of the structuring element varies depending on the level of the decomposition at which it is applied. The different sizes have been chosen satisfying a trade-off between ensuring that any possible interior region can be detected and that current regions do not generate seeds.

A three steps cleaning procedure has been presented for obtaining the final seed image from the result of applying the contrast extraction on error images. The first step removes all points in the seed image laying on the boundaries of the original segmentation regions. The second step is a thresholding devoted to the removal of small valued elements, which are usually related to textured areas rather than to non-detected interior regions. Finally, a cleaning step relying on the size of the elements is carried out. This final step removes small components of the seed image by performing a directional open filtering. In this way, components formed by one or two pixels are withdrawn, since they correspond very likely to textured areas.

Once the seed image has been obtained, two techniques have been analysed to introduce the seed information in the segmentation procedure. In the first case, seed information guides the algorithm to the locations within the image where new regions can be created. This technique arises problems when creating regions, since new regions contain only one pixel and, in such cases, parameter estimations cannot be carried out reliably. Furthermore, seeds can be split into more than one region in the final segmentation. This oversegmentation depends on the shape of the seeds as well as on the scanning technique.

The second technique introduces directly the seeds in the previous partition and reapplies the segmentation algorithm using this new partition as initial one. In this way, the above problems are overcome since new regions contain a fair amount of pixels and they are created at once. On the other hand, the direct creation of new regions may result in introducing erroneous regions in the initial segmentation. However, the algorithm is known to be able to remove such kind of regions. Therefore, this second technique has been chosen.

The above segmentation scheme has been tested on very different images. Several results obtained by means of this final segmentation scheme have been presented and discussed. Final results show clear improvements with respect to those presented in Chapter IV. Several details, which are not detected by means of the segmentation technique of Chapter IV, are extracted when using the seed information. Moreover, global segmentations of textured areas are not spoiled owing to the use of seeds for leading local analyses.