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 IV

MULTIRESOLUTION SEGMENTATION

In the previous chapter, a monoresolution segmentation technique has been presented. Although the method yields good results, it presents some drawbacks. These drawbacks are mainly produced by the fact that, in order to reduce its computational load, the algorithm carries out a maximisation procedure in a deterministic way. Therefore, the quality of the initial segmentation is critical to the quality of the final result. Actually, also the computational time required by the algorithm to achieve the final segmentation depends very much on the initial segmentation. Furthermore, as emphasised in Chapter I, the use of a monoresolution approach strongly constrains the performance of the method. Monoresolution techniques provide only with a very local viewpoint of images. Thus, segmenting correctly textured areas (either directly or starting from an oversegmented result) becomes rather difficult.

In order to overcome the above problems, a multiresolution approach is adopted. This chapter is devoted to the presentation of a multiresolution algorithm for segmenting still, gray level images. Although using a multiple resolution approach, the algorithm is based on the monoresolution method presented in the previous chapter. That is, the image is first decomposed into a set of images at different resolutions. In this work, the concept of resolution is related to that of frequency. At each level of the decomposition, and starting by the coarsest resolution, images are segmented using the monoresolution algorithm. The final segmentation at each level is utilised as initial segmentation for the next finer level of the decomposition. This procedure is performed through the whole decomposition up to the finest resolution level, which is the original image itself.

With such an approach, large textured areas can be detected as a few regions. This improvement is owing to the fact that the algorithm combines the information of the different resolutions in order to segment the image. Therefore, the oversegmenting effect is reduced. In addition, spurious regions are less likely to appear. Since every segmentation is performed assuming the same kind of image model, initial segmentations at each level are more consistent with the expected final segmentation. Finally, local maxima are easier to avoid when using a multiresolution approach. Solutions achieved at coarse resolution lead finer resolution steps to starting points closer to the actual optimum.

IV.1.- Multiresolution and compound random fields

Few works exist in the literature combining multiresolution analysis with compound random fields (CRF) based models. The first approach for using both concepts together is summarised in [67]. In this work, two segmentation methods are proposed, which can be applied isolately or together. The first method uses a multiresolution decomposition and regions are modelled by means of noncausal Gaussian Markov random fields (GMRFs). That is, no compound random field is used as image model. The multiresolution analysis is performed as in a "split & merge" approach. Regions are split into four square subregions and each region is assumed to contain samples from, at most, two different classes, out of N possible ones. The actual number of classes (N) must be given before hand. Each subregion is classified as belonging to one of these two classes by a maximum likelihood estimation. In order to account for the relation between neighbour regions, a

model for interactions between regions is defined. The procedure is iterated at each resolution until the pixel level is reached.

The second algorithm presented in [67] is basically a monoresolution technique which uses a deterministic approach for maximising the a posteriori likelihood of the segmentation, given the original image. In this case, images are modelled by a CRF. The upper level of the CRF is composed of a set of noncausal GMRFs whereas a Strauss process is assumed for the lower level. The Strauss process is defined on a second-order neighbourhood and non-zero potentials are assigned only to two-site cliques. These potentials are defined as it has been described in Section II.2.6; that is, $\beta_1 = \beta_2$. Each pixel within the image is assumed to belong to at most two possible classes, out of N. Hence, this assumption constrains the search of new possible solutions in the maximisation procedure.

The joint use of both approaches relies on the multiresolution technique for obtaining a good first segmentation and on the second algorithm for a posterior refinement of this segmentation. Moreover, the estimation of the parameters characterising the upper level of the CRF is also performed iteratively at each resolution. Note that the model of region used in the first approach corresponds to the model used in the upper level of the second method. On their turn, the values for the lower level parameters are set a priori.

Main drawbacks of this algorithm are rather clear. First, as it has been emphasised in previous chapters, a quadtree data structure does not totally exploit the advantages of a multiresolution analysis. In pure quadtree approaches, the concept of resolution deals only with the amount of data jointly analysed. Furthermore, the way of grouping data is rather simple (making square blocks) and does not take into account any characteristic of the image being analysed. A second problem of quadtree data structure is the little possibility for overcoming possible segmentation mistakes carried out at high levels of the data structure. That is, regions that have been split are no longer related to each other. Here, this problem is partially alleviated owing to the introduction of the model for interactions between regions which increases the importance of their relationship.

Another drawback of this algorithm is that the number of regions present in the final segmentation (N) has to be given a priori. Moreover, in order to simplify the algorithm, each block is assumed to be formed by samples from only two classes, out of the N possible ones. The last drawback, but not the

least, of this algorithm is that it does not really use jointly multiple resolution decomposition and CRF modelling. Actually, the decomposition is only used to provide with a good first segmentation and good parameter estimates to a second algorithm, which further improves this segmentation by a MAP criterion involving CRF modelling.

A similar work is discussed in [68], showing the algorithms the same sort of problems than in [67]. Here, the concept of uncertainty is introduced in the first segmentation stage, since regions of the image can be initially classified as ambiguous. Ambiguous regions are not used for estimating the model parameters. In addition, the performance of the final refinements on the initial segmentation is studied with both deterministic and stochastic techniques. The final segmentation quality yielded by both techniques are very similar, stochastic techniques being much more computationally demanding. Analogous results have been presented in Chapter II, being the reason for this similarity in performance the fact of driving deterministic algorithms with good initial segmentations.

In [69], another segmentation algorithm combining multiresolution analysis and image modelling by means of CRFs is developed. The algorithm is regarded as an extension of the basic K-means clustering algorithm to account for spatial constraints and intensity variations within regions. Thus, some information about the number of classes of the final segmentation must be provided to the algorithm. The lower level of the CRF is defined as in [67], while the model for each region is a slowly varying intensity function plus a white Gaussian noise. The segmentation is carried out by maximising the a posteriori probability using a deterministic algorithm. This task and the estimation of the intensity functions are interleaved. Multiresolution analysis is introduced in order to obtain robust estimates of the model parameters. Initially, intensity functions are assumed to be constant and their parameters are estimated by averaging, at each point, the data within a sliding window. The estimation is performed in a top-down fashion, since as the algorithm progresses, the size of the window reduces. Therefore, estimations are initially global and progressively adapt to local. Finally, in order to improve the final segmentation, a merging step based on the mean gray level difference calculated along the border of neighbour regions is proposed.

As in the previous works, the algorithm presented in [69] only uses the multiresolution analysis for obtaining good model parameter estimates. That is, the segmentation itself is always carried out at the same resolution. However, in [69] the concept of resolution is expanded. It copes not only with

the idea of the amount of data to be jointly used but also with the concept of filtering. At each resolution a filter of different size is used, which leads to a different discrimination in the frequency domain.

It has to be noticed that the algorithm of [69] is mainly a clustering algorithm (extension of the K-means algorithm) whose results are used for segmentation. Hence, the necessity of providing initially with the number of classes. Furthermore, in order not to make too slow the algorithm, only a few classes are allowed. Results presented in [69] cope with the classification of images, mainly formed by flat regions, into only 2, 4 or 8 classes. Even with such simple regions, the algorithm cannot perform the segmentation in a top-down way and needs a final bottom-up stage for merging adjacent regions.

A technique using multiresolution approaches for both parameter estimation and image segmentation is presented in [70]. The image model is a CRF whose upper level is a set of causal nonhomogeneous Gaussian random fields. The reason for choosing such a model, in spite of its causal assumption, is that the estimation of its parameters is a feasible task. On its turn, the segmentation field (low level) is modelled by a second order Strauss process. Potential values are defined as in Section III.1.2; that is, only two-pixel cliques are considered and their potential definition takes into account clique homogeneity. Their actual values are set by assuming that horizontal, vertical and diagonal boundaries should yield equal energy functions.

The parameter estimation technique presented in [70] can be seen as an extension of that presented in [68] to obtain the number of textures within the image, in addition to their parameters. Images are split into several square blocks. Each block is coarsely classified following a criterion based on their estimated mean and variance. Model parameters are estimated from each cluster. A cost for merging clusters is defined and all mergings with negative cost are performed. After finishing all the possible mergings, the cluster partition is recomputed. The clustering refining procedure is iterated until no change in the partition is introduced. This procedure is analogous to that of other algorithms such as the K-means. The cluster partition results in a set of estimated parameters which is used to carry out a coarse segmentation of the image. Image blocks split by segmentation boundaries are assumed to contain data from more than one field and, therefore, are removed from the parameter estimation procedure. Finally, model parameters are recalculated with the remaining blocks and the cluster merging procedure is applyed again in order to obtain the actual number of classes.

Once the parameters of the model have been estimated, the segmentation is carried out following a multiresolution approach. A decomposition in multiple resolutions is performed following a quadtree data structure. At each resolution, a deterministic segmentation is carried out assuming that the lower level parameters remain equal through the different resolutions. In order to formulate the likelihood functions for each level, a method for decimating likelihood functions is proposed. The segmentation obtained at a given resolution is used as starting point in the segmentation procedure of the next finer resolution. Therefore, the segmentation procedure is a mixture of bottom-up (for computing the likelihood functions) and top-down approaches (for performing the segmentation itself).

Results presented in [70] show the great improvement achieved by using multiresolution approaches. Segmentations obtained by monoresolution techniques relying on stochastic maximisation criteria (simulated annealing) are compared with those yielded by the multiple resolution segmentation (MRS). It can be seen that both techniques perform very similar, MRS requiring two orders of magnitude less computation (in average).

The work in [70] represents a big step towards the obtaining of a general purpose, well-behaved segmentation algorithm. The use of CRFs for image modelling jointly with a good exploitation of multiresolution analyses allows the algorithm to achieve good results in a feasible amount of time. Furthermore, since the algorithm estimates not only the parameters of the different models but also the number of models within the image, the technique is totally unsupervised.

However, some points in the algorithm should be further discussed. The algorithm makes a better use of the multiple resolution decomposition than the former methods, since it uses this approach for both parameter estimation and segmentation. Nevertheless, the algorithm carries out both tasks separately. Moreover, a different concept of resolution is used for performing each one of these tasks. When estimating parameters, the quadtree structure is used only for gathering the data to be jointly analysed, while when segmenting, the quadtree involves a decimation procedure. Therefore, two different data structure are necessary.

Another drawback of the algorithm is that it makes the assumption of having the same lower level parameters and number of classes for the whole multiresolution structure. This assumption allows the algorithm to perform the estimation of the number of classes only once. However, since a decimation

procedure is performed in order to obtain the multiresolution decomposition, this assumption seems far from real. Note that decimation can remove from the coarsest resolutions small regions present in the finest resolutions. Thus, some initial classes may not be present in all resolutions. In addition, boundaries of regions at different resolutions may not behave equal. Therefore, the parameters characterising their behaviour should be different.

Regarding the computational time required by the algorithm, although it leads to a great reduction with respect to stochastic based techniques, some improvements may be introduced. The initial computation of the model parameters uses an iterative procedure, which is very similar to the K-means algorithm. It should be highlighted that such a kind of algorithms are very time demanding. Moreover, in the results shown in [70], region interiors are characterised by 5 different parameters, which need a rather complex estimation technique. As it has been shown in Chapter III, simpler models (white GRFs) can be used yielding similar results while decreasing the computational load.

In addition, given that parameter estimation and segmentation are performed separately, the computation of two different multiresolution decompositions is necessary. Furthermore, the segmentation is neither a top-down nor a bottom-up pure approach. On the contrary, it makes use of a mixture of both approaches: bottom-up for decimating the likelihood functions and top-down for actually performing the segmentation. To have two different data structures and to carry out passes in both senses through one of the decomposition slow down the segmentation procedure.

Recently, a refinement on the previous work has been presented [71]. In it, the drawback of using a quadtree structure is partially overcome by making use of a linked pyramid data structure. However, there is not a simple recursion for obtaining the likelihood functions at the different levels of a linked pyramid. Due to this lack, a hybrid data structure is proposed. The data structure is a mixture between a linked pyramid for the coarsest levels of the decomposition ($n > 2$) and a quadtree for the finest levels ($n \leq 2$). The computation of the likelihood functions at the coarsest levels is approximated by a heuristic recursion. Hence, two different procedures for decimating likelihood functions are required, one for each kind of data structure. A second improvement deals with the parameter estimation technique. The algorithm presented in [71] utilises the expectation-maximisation algorithm [72] which is said to achieve better estimation results than the former criteria. Nevertheless, in order to estimate the model parameters, two different passes through the

whole data structure are necessary, each pass being a bottom-up followed by a top-down analysis.

As it has been emphasised when commenting the previous works, a very important feature of a multiresolution segmentation algorithm is the multiresolution decomposition itself. Several meanings can be given to the concept of resolution. In the above works, this concept is mainly related to the amount of data to be jointly processed [67, 68, 70]. However, this concept can be associated to other meanings such as frequency [69] or object size [73]. In the following section, the multiresolution decomposition that has been chosen in this work is presented, and the reasons for its selection discussed.

IV.2.- Multiresolution decomposition

In this work, the chosen multiresolution decomposition is a Gaussian Pyramid (GP) [74]. In this decomposition, the concept of resolution is related to that of frequency. That is, each resolution contains a replica of the original image from which some high frequencies have been removed. This removal is performed by low-pass filtering the image. As the level in the decomposition increases, the band-pass of the filter decreases and so does the resolution on the image. Assuming that, after filtering, the signal guaranties Nyquist theorem for signal subsampling [75], filtered images can be decimated. Therefore, the concept of resolution copes also with that of image size. In the sequel, the main features of a Gaussian pyramid decomposition are shortly overviewed.

IV.2.1.- The Gaussian pyramid

In a Gaussian pyramid, the original image is taken as the level (0) of the decomposition; that is, the bottom of the pyramid. For the remaining levels, the data within a level (l) are obtained by filtering the data within the previous level (l-1). Low-pass filters utilised for such a procedure are assumed to cover half of the band of the signal. Therefore, the image can be decimated, after filtering, by a factor of 2 in each spatial component. That is, if the image at level (l) has dimensions NxN, the resulting image at level (l+1) will have dimensions N/2xN/2. The pyramid structure is built by the layered arrangement of these successive smaller replicas of the original image. A block diagram illustrating the way to produce the Gaussian pyramid is shown in Figure IV.1.

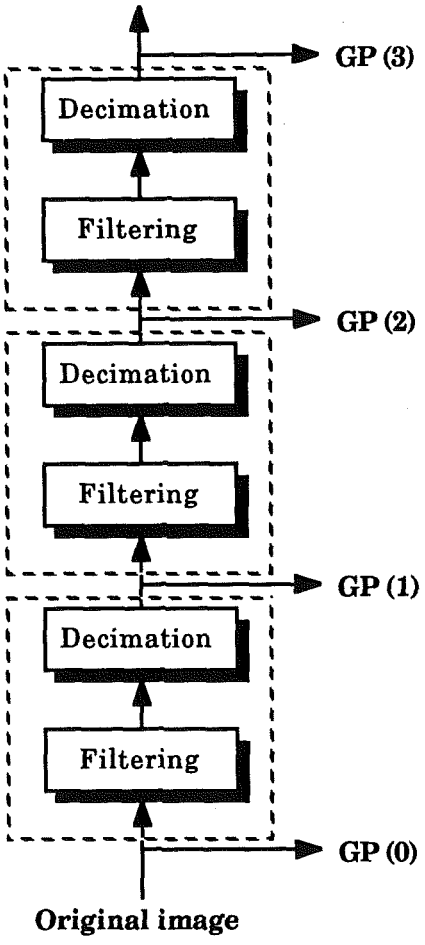

Fig. IV.1.- Block diagram of the procedure for creating the Gaussian pyramid

The decimation procedure allows applying, at each decomposition level, the same low-pass filter [74]. Different filter weight configurations, w (m, n), have been studied, mainly following the constraints proposed in [74]:

• **Filter length**: it has been seen that the size of the filter is not critical. However, a 3x3 filter may be too small and lead, in some cases, to strong aliases. Hence, a 5x5 filter has been chosen since it provides adequate filtering at low computational cost. The procedure for generating the Gaussian pyramid in such a case is shown in Figure IV.2 for a monodimensional example.

• **Separability**: in order to simplify the computations, the filter is chosen to be separable. That is,

$$w(m, n) = w_x(m) * w_y(n). \qquad \text{(IV.1)}$$

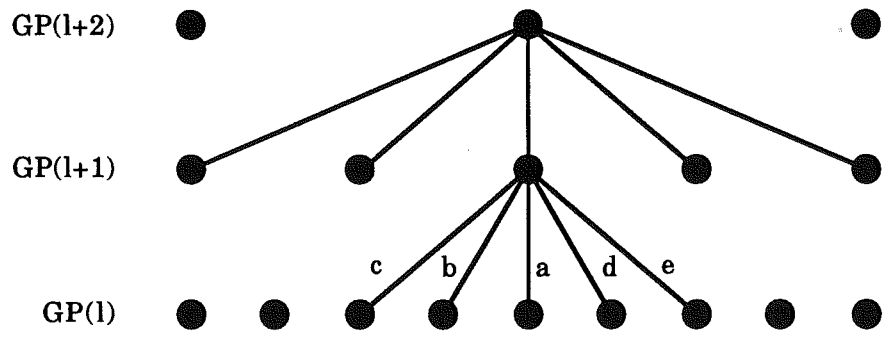

Fig. IV.2.- Graphic representation of the generation of a Gaussian pyramid

• **Normalisation**: the weights of each one of the monodimensional filters are set to sum 1:

$$\sum_{m=-2}^{2} w_x(m) = \sum_{n=-2}^{2} w_y(n) = 1 . \qquad \text{(IV.2)}$$

This normalisation is performed for preserving the mean value of the signal, after filtering.

• **Symmetry**: In order not to give priority to any spatial direction, both filters are assumed to be equal and symmetric with respect to its central point. Therefore,

$$w_x(k) = w_y(k) = w(k) \qquad \text{(IV.3)}$$

and

$$w(k) = w(-k) \quad \text{for } k = 0, 1, 2. \qquad \text{(IV.4)}$$

Assuming the coefficients given in Figure IV.2, (IV.4) translates into:

$$w(0) = a, \quad w(1) = w(-1) = b, \quad w(2) = w(-2) = c . \qquad \text{(IV.5)}$$

• **Equal contribution**: This constraint stipulates that all the points at a given level (l) should contribute the same to the following level (l+1). This constraint results in having a single weight characterising the whole filter (shape parameter):

$$w(0) = a, \quad w(1) = 1/4, \quad w(2) = 1/4 - a/2 . \qquad \text{(IV.6)}$$

The frequency behaviour of the filter given by (IV.6) – W (f) – with respect to the shape parameter can be seen in Figure IV.3. Note that the range of values in

which the filter is unimodal (that is, closer to a Gaussian filter) is a ∈ [0.4, 0.6]. Actually, only in the case a = 0.4 the filter can be said to be Gaussian.

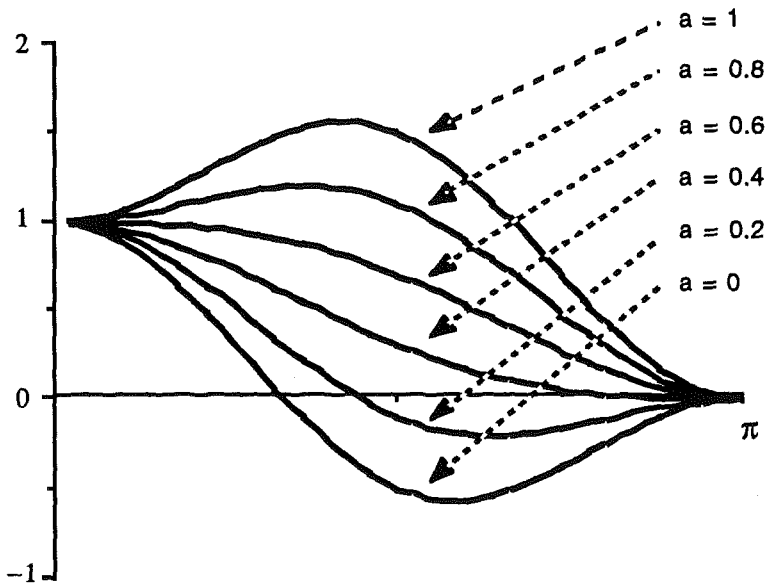

Fig. IV.3.- Frequency response of the filter when varying the shape parameter

IV.2.2.- The Laplacian pyramid

There exists a second pyramid which is very related to the Gaussian pyramid. This pyramid is called Laplacian Pyramid (LP) and stores the information which is necessary for going from the level (l+1) of the Gaussian pyramid to its following finer resolution level (l). That is, the Laplacian pyramid keeps, at level (l), the difference between two consecutive levels of the Gaussian pyramid. Given that each Gaussian pyramid level is a low-pass version of the original image, the Laplacian pyramid is formed by a set of band-pass versions of the original image.

Note that the difference between two consecutive levels of the Gaussian pyramid cannot be computed directly, since these images have different sizes. In order to compute this difference, a previous interpolation of the level (l+1) of the Gaussian pyramid has to be carried out. Usually, the filter applied in this interpolation is the same filter that has been used for computing the different levels of the Gaussian pyramid [74]. A block diagram illustrating the procedure for obtaining both pyramids is shown in Figure IV.4. At each level, the main operations have been gathered within a dashed block. In the sequel,

when illustrating any procedure dealing with the pyramid construction, this single block will represent the whole computation of both pyramids.

Regardless the kind of filter used in the interpolation, this decomposition ensures the perfect reconstruction of any level of the Gaussian pyramid. Of course, attention must be paid to using the same filter for interpolating when building the pyramids and when computing the reconstruction. The only necessary information for this reconstruction is the top level of the Gaussian pyramid and the whole Laplacian pyramid down to the level whose reconstruction is sought.

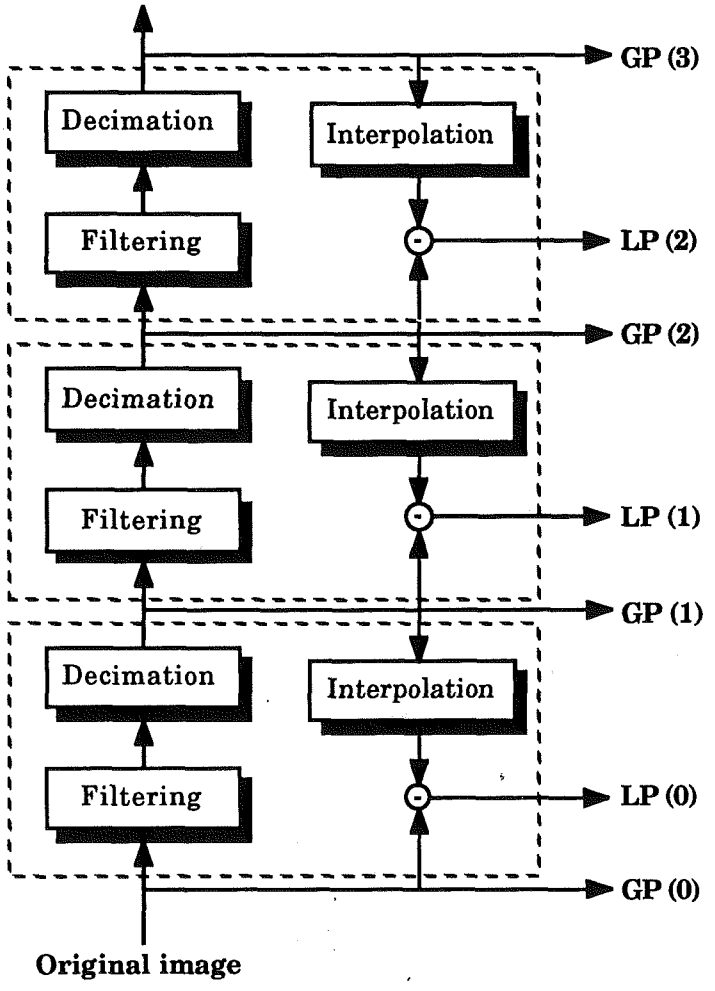

Fig. IV.4.- Block diagram of the procedure to obtain the Gaussian and Laplacian pyramids

An example of the decomposition of an image into its Gaussian and Laplacian pyramids is shown in Figure IV.5. It has to be noticed that, since images composing the Laplacian pyramid come from a difference between two

Gaussian pyramid levels, the values of their pixels are not necessarily positive. In order to represent the levels of the Laplacian pyramid as normal images, the absolute value of each pixel is represented. Moreover, Laplacian images have typically very small values. Therefore, to enhance them, each level has been separately scaled.

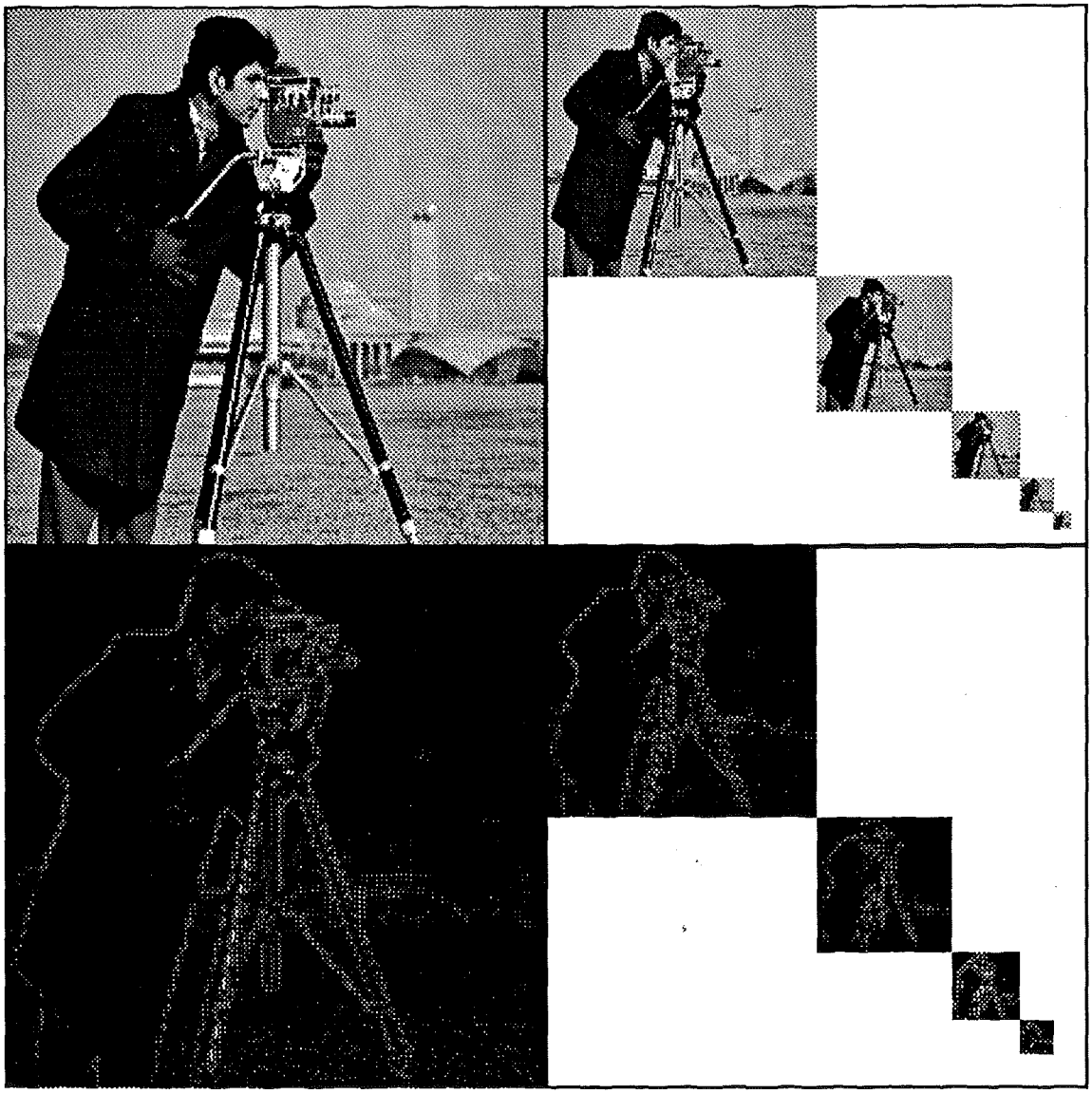

Fig. IV.5.- Gaussian and Laplacian pyramids of the Cameraman image

The shape parameter used for building the Gaussian pyramid of Figure IV.5 has been set to 0.6. Note that the information contained in a Laplacian pyramid is very related to contours and details within the image. Thus, this information may be important in order to control the segmentation procedure, as it will be seen in the following sections.

IV.2.3.- Comments on the choice of the decomposition

Multiresolution analysis is introduced in the segmentation algorithm presented in Chapter III so that the main problems arisen by this monoresolution algorithm can be overcome. As it has been pointed out at the beginning of this chapter, the most of these problems are related to the possible poor quality of the initial segmentation. As the algorithm performs a deterministic maximisation, it can very likely get trapped in local maxima. For instance, final results may present oversegmented textured areas or spurious regions, depending on the initial segmentation. These drawbacks can be overcome by using the segmentation of coarse levels of the decomposition for leading the segmentation of the fine levels.

The use of a Gaussian pyramid as multiresolution decomposition results in having a sorted set of low-pass filtered, reduced replicas of the original image. Low-pass filtering smooths the information within images by removing its rapid fluctuations. On its turn, decimation reduces the image size. Thus, texture information, which may be spread over large areas in the finest resolution, gathers into a few pixels at coarse resolutions. It is worth noticing that, regardless the statistics of the initial data, the statistics of coarse resolution data are closer to Gaussian statistics than fine resolution ones, due to the filtering procedure. This statement is based on the Central Limit theorem [41]. Therefore, textures at coarse resolution are closer to the assumption of Gaussianity which is made by the monoresolution segmentation algorithm and, thus, their detection becomes easier.

The algorithm presented in Chapter III makes use of an initial segmentation in order to obtain the final segmentation. Here, the segmentation of a level is used as rough estimation of the segmentation of the following level. Rough segmentations can be easily performed at coarse levels of the Gaussian pyramid decomposition. This ease is mainly twofold. First, low-pass filtering removes details and small regions from the finest level images. Thus, data in coarse levels are simpler and more homogeneous. Second, the size of high levels on the pyramid is smaller than that of the low levels. Hence, high level segmentations involve many less pixels and can be performed faster than low level ones.

In any multiple resolution decomposition, similar features are kept at the same decomposition level. Therefore, analyses performed at each resolution level can handle data with homogeneous features. In a multiresolution segmentation, this property allows tuning the image model to

characterise the current data, at each level. This tuning possibility should be related to the kind of image model and the concept of resolution which are assumed. Therefore, in this case, the tuning is related to the importance of the lower level model with respect to the upper level, since, due to the filtering effect, their relation varies through the levels of the Gaussian pyramid.

The chosen multiresolution decomposition not only provides with the information contained in the Gaussian pyramid, but also with that related to the Laplacian pyramid. In this second pyramid, information about the difference between two consecutive levels of the Gaussian pyramid is stored. Therefore, this information can be used for leading the refinements that have to be carried out in the initial segmentation in order to obtain the final segmentation of each level.

On the other hand, it should be pointed out that images at coarse levels of the Gaussian pyramid are blurred, due to the filtering process. This blurring results in an uncertainty in the position of the boundaries of the regions. Therefore, contours obtained at coarse resolutions may not exactly conform those contours related to the same regions at fine resolutions. However, this effect is not a drawback when leading fine level segmentations by coarse level ones. It should be recalled that the segmentation procedure at each decomposition level is mainly performed by refining the boundaries of the initial segmentation. Thus, right boundaries at each level are finally obtained.

A more detailed study of the characteristics of this decomposition when used as basis for a segmentation algorithm is performed in the following sections. Several experiments have been carried out in order to choose the best low-pass filter for creating the Gaussian pyramid. The set of constraints proposed in [74] have been selected. That is, the chosen filter is of size 5x5, separable, normalised, symmetric and with coefficients having equal contribution. Regarding the shape parameter, the best segmentation results have been obtained by setting a = 0.6. Therefore, this value will be used in the sequel.

IV.3.- Supervised multiresolution segmentation

The multiresolution segmentation algorithm, that has been outlined in the previous section, is developed in this section. The base of the algorithm is the use of the monoresolution segmentation technique presented in Chapter III, at each level of the Gaussian pyramid. The initial segmentation at each level is provided by the segmentation of the previous coarser level. This procedure is iterated down to the finest level of the decomposition is segmented. Given these general concepts, some points on the algorithm deserve further discussion.

IV.3.1.- Coarsest level segmentation

The segmentation procedure starts by segmenting the image at coarsest resolution; that is, the top of the pyramid. Since no initial segmentation is available for this level, the algorithm has to obtain it. Typically, images at this level are small (32x32, 16x16 or even 8x8 pixels). Therefore, the segmentation at this level could be performed by using any high-quality, time-demanding segmentation algorithm without implying a large computational effort. Nevertheless, it has been seen that nearly any kind of segmentation technique yields similar results when applied to such images. This property is owing to the special characteristics of this kind of images. Note that, in addition to the fact of being small, images at the top level of the pyramid share the feature of being the result of a strong low-pass filtering. Thus, almost no kind of homogeneity criteria can be applied to such images, rather than a gray level homogeneity criterion.

Given the above results, the simplest initial segmentation has been chosen in order to start the segmentation procedure. That is, for the image at the top of the pyramid, an initial segmentation in which each pixel is a single region has been assumed. The algorithm described in Chapter III is then applied in order to achieve the final segmentation at this level.

Regarding the necessary number of levels in the pyramid, a study has been carried out analysing the trade-off between the computational time required for building the pyramid and the quality of the yielded results. Note that the smaller the number of levels, the closer the algorithm to the monoresolution approach. Segmentation procedures starting at levels higher than level four (l = 4) turn out not to improve final results. Actually, final segmentations at level five and beyond seldom present regions with more than one pixel. On the other hand, starting the procedure at level four does improve

the quality of final results, while increasing very little the computational load. Thus, Gaussian pyramids have been chosen to be formed by five levels; that is, they are built up to level four (l = 4) since the bottom level (the original image) is denoted by (l = 0).

IV.3.2.- Partition interpolation

Once a level (l) of the Gaussian pyramid has been segmented, this result is utilised as initial point for the segmentation of the next finer level (l-1). In order to apply the algorithm of Chapter III to this pyramid level, both images (the initial segmentation and the original image) should have identical dimensions. It should be noticed that, since the initial segmentation provides from the previous level of the pyramid, its dimensions are half the dimensions of the image to be segmented. Hence, the necessity of interpolating the previous partition.

The interpolation of a partition can be seen as a contour interpolation problem. Several contour interpolation techniques have been tested. Sophisticated methods, such as splines or similar, are absolutely out of question. Such techniques are avoided given that, contours at a level (l) may not correctly conform contours at the following finer level (l-1). Thus, simple techniques relying on the replica of pixels have been chosen. Beside the trivial solution of taking a square block of four pixels as interpolation pattern, other patterns have been tried in order to prevent the apparition of block-like contours (see Figure IV.6).

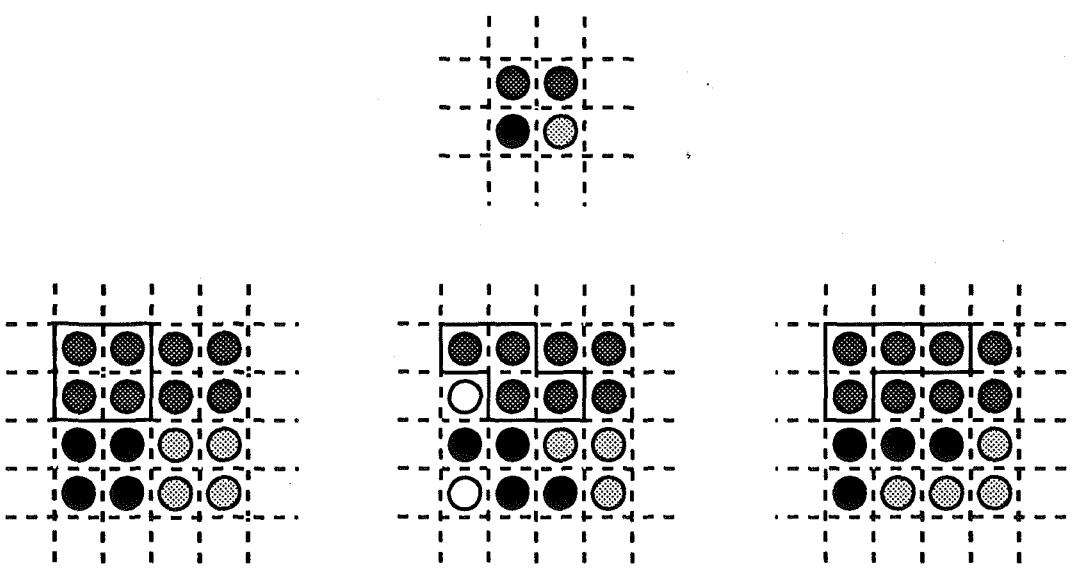

Fig. IV.6.- Initial partition and three different kinds of interpolation pattern

Block-like contours could arise some problems since, regarding the image model, these contours lead to very low energy function values and, therefore, to very high likelihood function values (see Chapter III). Hence, the initial contour configuration may represent a local maximum or be very close to one. However, results obtained with the whole set of interpolation patterns are very alike and the algorithm does not get trapped in such kind of local maxima. Therefore, the square block solution is used, given that it allows a fast implementation.

IV.3.3.- Basic structure

In Figure IV.7, a block diagram of the basic structure of the multiple resolution segmentation algorithm is shown. This basic structure was first presented in [76]. The Segmentation blocks correspond to the monoresolution segmentation algorithm presented in Chapter III.

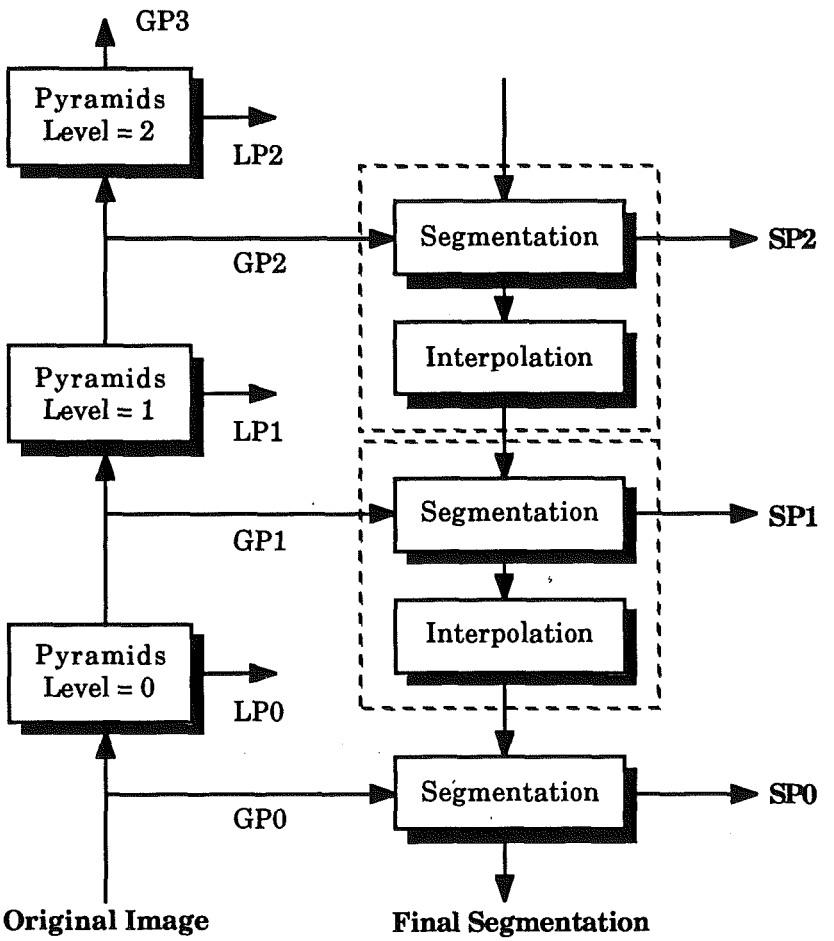

Fig. IV.7.- Block diagram of the multiresolution segmentation basic structure

The Segmentation and Interpolation blocks have been grouped into a single block (dashed in the diagram). In the sequel, when illustrating any procedure dealing with the segmentation of a level of the pyramid, these two blocks will be represented as a single one, unless referring to level 0. Note that the interpolation at level 0 is not necessary, since this is the last level to be segmented. As a segmentation is performed at each level of the decomposition, a third pyramid can be built by storing the different level segmentations. In the block diagram this pyramid has been denoted as Segmented Pyramid (SP).

At each level of the decomposition, a different maximisation is carried out. The likelihood function which is maximised can be written as an extension of (III.5):

$$P(Q(l) = q(l), X(l) = x(l)) = \frac{1}{Z} \exp\left(-\frac{1}{T^*(l)}\left[k_1(l) + k_2(l)\, V^*(l)\right]\right) \cdot$$

$$\cdot \prod_{R_n(l)} \frac{1}{\sqrt{(2\Pi)^N}\, \sigma(l)_n^N} \exp\left\{-\frac{\sum [x(l) - \mu_n(l)]^2}{2\,\sigma(l)_n^2}\right\}, \qquad \text{(IV.7)}$$

where all the parameters are defined as in Chapter III and (l) denotes the level in the decomposition. Note that the use of a multiresolution approach opens the possibility of giving different values to the model parameters at each level.

Actually, given the structure of the monoresolution segmentation algorithm, this possibility does not lead to any improvement regarding the texture parameters. Note that the parameter values of the upper level model are constantly updated. Therefore, the model not only conforms to the data at each level but to the partition at each moment. However, to change the values of the parameter controlling the importance of the contour model with respect to the texture (T*) as well as the parameter controlling the contour shape (V*) may result in some improvements within the segmentation method.

The use of different values of V* does not lead to any improvement. As it has been seen in Section III.3.3, segmentations differing only in the value of V* are of very similar quality. In addition, varying V* does not change the computational load of the whole process. Therefore, almost any value of V* could be used, at each level, for obtaining good segmentations. However, it has also been emphasised that, by assuming $V^* \in [0.25, 1]$, final contours are smooth and conform to natural boundaries. Thus, at the bottom of the pyramid (l = 0), a potential ratio belonging to the above set should be used. Since for the

rest of the levels there is not a clear constraint, it seems natural to use for the whole pyramid the same parameter value. In this way, V* (l) = V* = 0.5 has been selected.

The possibility of varying the importance of the contour model with respect to the texture model through the decomposition is briefly discussed in [70]. Two different approaches are outlined in [70]. The first approach consists in fixing the value of the energy function of contours through all levels. That is, given a contour at the coarsest level, it must have associated the same energy function value as all the contours obtained, by its interpolation, at each level of the decomposition. This approach leads to image models in which contours are less important with respect to textures at the finest levels than at the coarsest levels. Due to this property, this approach is withdrawn and the assumption of not having any a priori information is taken. That is, the same model for contours is used through the whole decomposition.

IV.3.4.- Study of the performance of the basic structure

In the case of using a Gaussian pyramid, to fix the value of T* for the whole decomposition is not a realistic approach. Due to the filtering procedure, the importance of textures with respect to contours varies through the decomposition. In spite of knowing that, this assumption has been taken in order to analyse the performance of the basic structure above presented. This analysis, as in Chapter III, has been mainly performed using the image Cameraman. In Figure IV.8 a set of final segmentations obtained by means of this basic structure is shown. The different results have been achieved by changing the value of T*, which remains fixed for the whole segmentation. These values, as well as the final number of regions, are shown in Table 1.

(i, j)	(0, 0)	(0, 1)	(0, 2)	(1, 0)	(1, 1)	(1, 2)	(2, 0)	(2, 1)	(2, 2)
T*	$\rightarrow 0$	0.2	0.3	0.8	1	1.2	3	100	$\rightarrow \infty$
# Reg.	9	39	80	155	174	233	520	806	7254

Table IV.1.- Data from segmentations of Figure IV.8

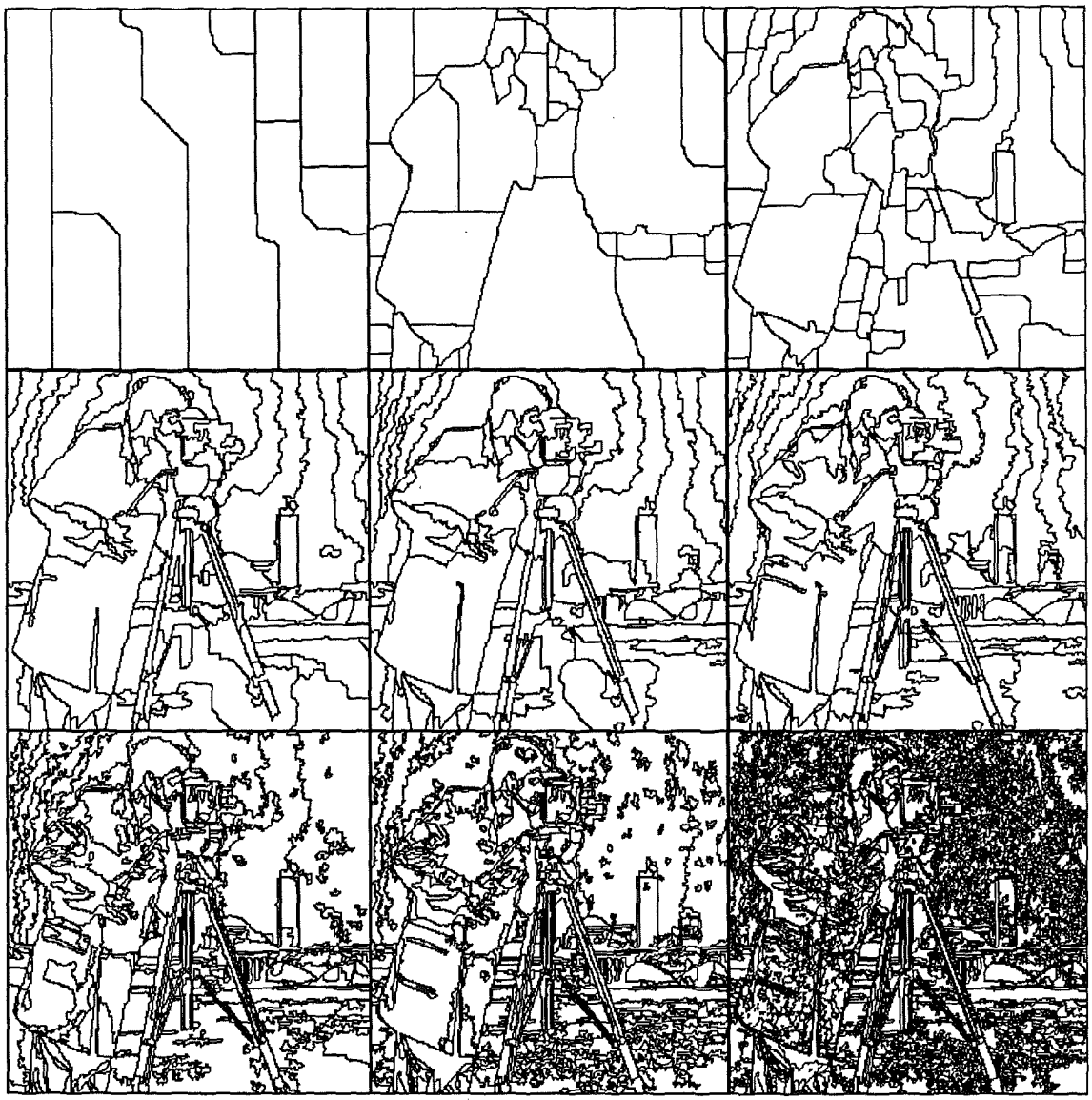

Fig. IV.8.- Segmentations obtained with the multiresolution basic structure

Several points have to be highlighted from the above data. First, it is worth noticing that the multiresolution algorithm is not constrained by the initial segmentation at the top of the pyramid. When applying the monoresolution algorithm with extreme values of T^* ($T^* \rightarrow 0$ or $T^* \rightarrow \infty$), final segmentations are rather alike (see Figure III.14). As it has been discussed in Section III.3.2, this effect is owing to the fact that deterministic maximisation techniques are very depending on the starting point. That is, given the initial segmentation, the possible final results are very constrained.

A multiresolution approach overcomes, almost totally, this problem. This improvement can be seen in the results obtained when using $T^* \rightarrow \infty$ and $T^* \rightarrow 0$. In both cases, results are quite close to the expected maxima of the

likelihood functions (that is, each pixel representing a region and the whole image being a single region, respectively). In Figure IV.9, a graphic illustrating the final number of regions achieved when varying the parameter T* is shown. Note that for large values of T*, as well as for small ones, the final number of regions remains nearly constant.

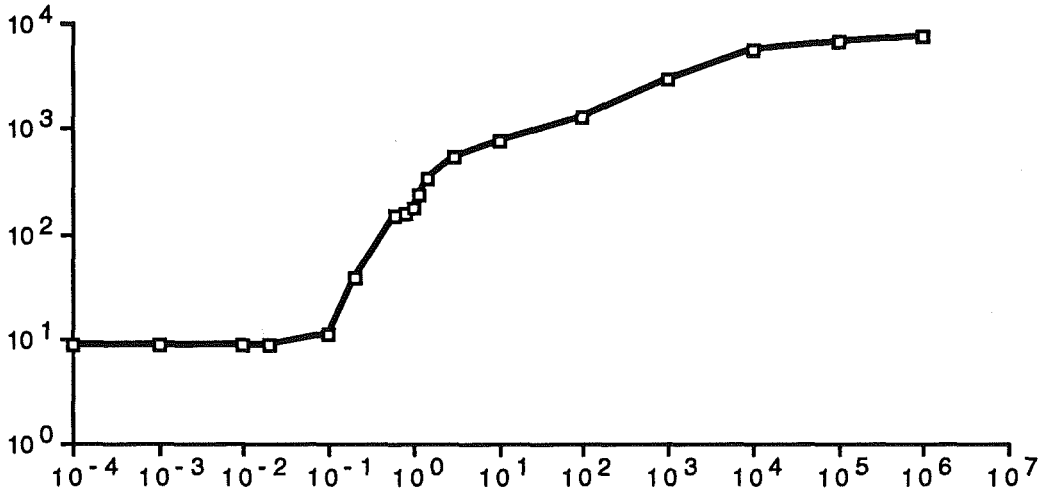

Fig. IV.9.- The number of regions in the final segmentation is plotted versus the value of T*

Although a large set of different segmentations can be achieved by changing the value of T*, good quality segmentations are yielded by fixing this parameter in a range of values around T* = 1. Note that a similar result has been obtained in Chapter III for the monoresolution algorithm. It is worth emphasising the algorithm performance within this range of values when segmenting textured areas. This feature can be observed in the segmentation of the grass in the middle examples of Figure IV.8. Here, this textured area is segmented into a few regions, unlike in the examples presented in Chapter III where a large amount of regions was used.

This improvement is owing to the use of a multiresolution approach. In order to illustrate this statement, Figure IV.10 shows the original Cameraman image, the segmentations of each level of its Gaussian pyramid and the final segmentation as a contour and as a mosaic image (174 regions). This segmentation has been obtained by setting T* = 1. It can be seen that the grass is correctly segmented from the coarsest levels of the pyramid, and that the use of these segmentations in order to guide the finest levels has made possible the final segmentation. The algorithm capability for coping with textured areas will be further illustrated in this section.

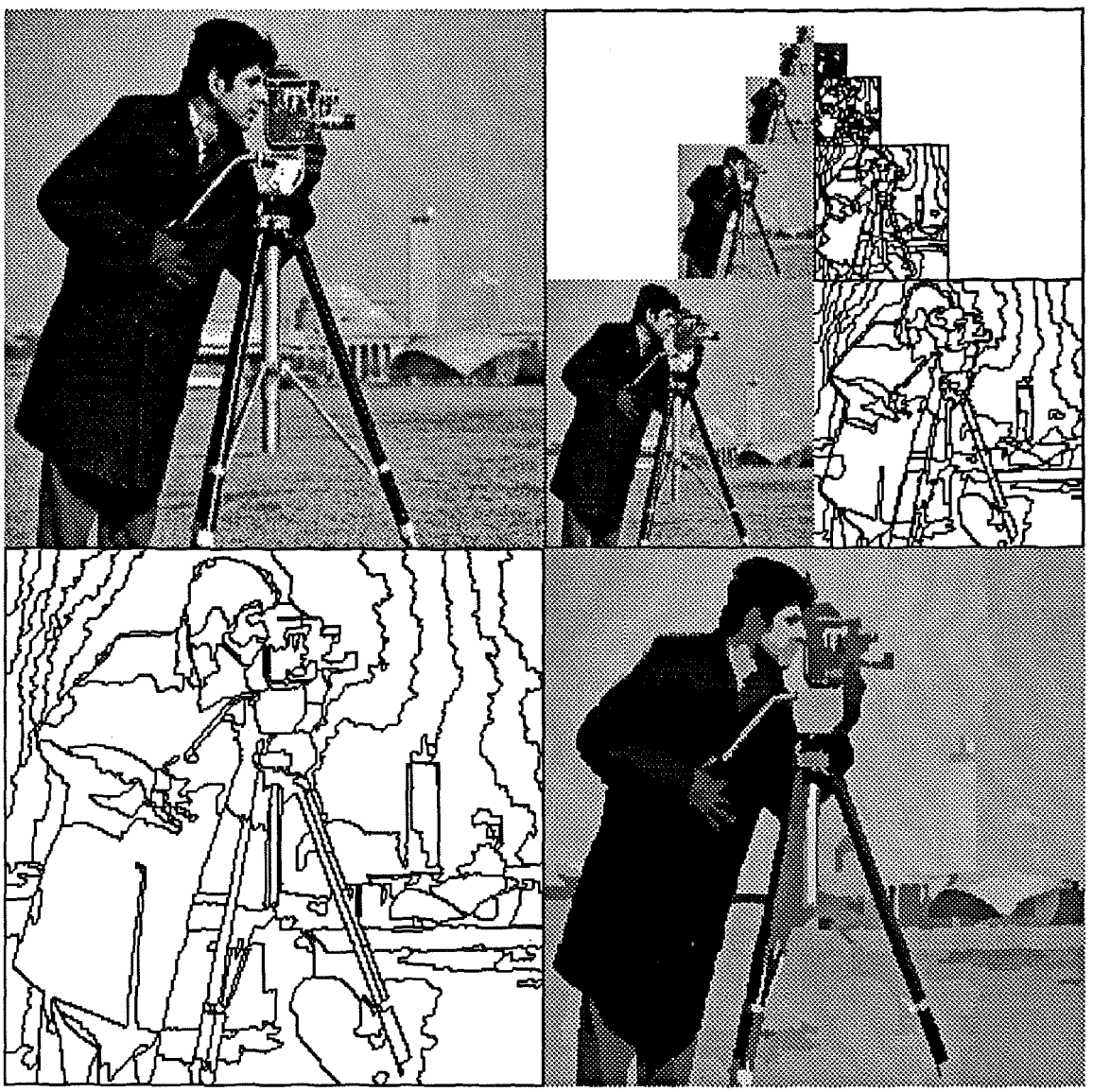

Fig. IV.10.- Gaussian and Segmented pyramid of Cameraman

Another point to be highlighted concerns the computational load. The multiresolution algorithm implies a much lower computational load than the monoresolution algorithm. Although the total number of scans necessary to obtain the segmentation increases (a segmentation has to be performed at each level of the Gaussian pyramid), the computational load does not. The reason for this decreasing is twofold. First, in the monoresolution algorithm, the amount of time devoted to the computation of the initial segmentation has to be added to that of the algorithm itself. As it has been discussed in Chapter III, in order to have good initial segmentations, complex algorithms are required.

Second, in the multiresolution algorithm, a large amount of scans are carried out at the coarsest levels of the decomposition. Given that images at coarsest levels are smaller than the original image, these scans are performed faster. Once the main structure of the image has been detected in the coarsest levels, segmentation at finest levels only requires contour refinements which can be performed in few scans. Typically, only two scans are necessary at the finest level in order to obtain the final segmentation.

On the other hand, this basic multiresolution structure arises some drawbacks. Details in the image are not correctly segmented and some small regions have been removed. These effects can be seen in the different results presented in Figure IV.8 as well as in Figure IV.10. Note that small details in the camera or the face of the man are not well preserved and neither are some parts of the tripod. There are mainly two reasons for having such results. The algorithm uses the same value of T* at all levels of the decomposition, regardless the kind of information contained in them. That is, the same image model is assumed for each level of the pyramid. This assumption leads to wrong results at coarse levels, which guide to inaccurate segmentations at fine levels. The second reason is related to the problem of detecting interior regions, discussed in Section III.3.4. That is, the algorithm is not able to detect interior regions, given that the segmentation is performed by a continuous contour refining.

The second main drawback of the basic structure is that, although good segmentations are obtained for a large range of values of parameter T*, the best results for each image are not achieved by using the same value. In order to illustrate this point, Figures IV.11 and IV.12 show six different examples of image segmentation. Images have been grouped into two classes depending on the amount of textured area that they contain. In this way, images in Figure IV.11 are mainly not textured while those in Figure IV.12 present large textured areas. In these figures, original images and best segmentations obtained for each case (as contour and as mosaic images) are shown. In order to choose them, the algorithm has been carried out using values of $T^* \in [0.4, 4]$ with increments of 0.2. The quality criteria have been both objective (final number of regions) and subjective (visual quality of the mosaic image). In all cases, segmentations present similar achievements and drawbacks as in the Cameraman case.

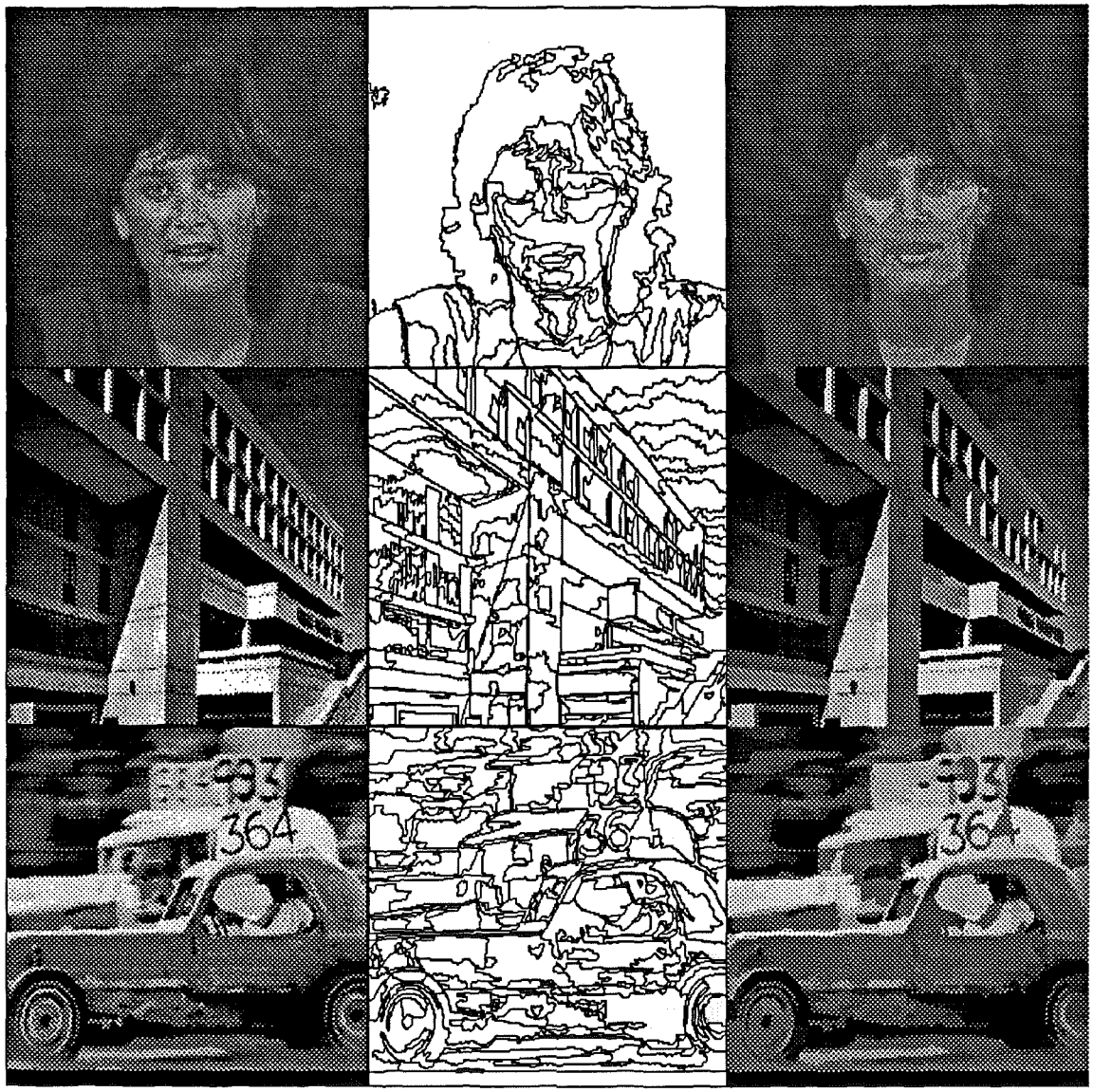

Fig. IV.11.- Examples of segmentations of little textured images

In the first row, segmentation is performed on the Miss America image (T* = 1.2, 131 regions). Note that the background of the image has been successfully gathered into a single region not overlapping with the hair. However, some details have not been correctly segmented, as the left eye, or have not been detected at all, as the teeth or the nose. In addition, a small spurious region appears within the background (left side of the image). The result of segmenting the Building image is shown in the second row (T* = 1.4, 376 regions). Although some windows at the top of the building have been overlooked, the good quality of this segmentation should be highlighted. The third row is devoted to show the segmentation of the Cars image (T* = 1.2, 397 regions). As in the previous case, the quality of this segmentation has to be

emphasised, in spite of the loss of some details (e. g.: the numbers of the cars or the central black spot of the right wheel).

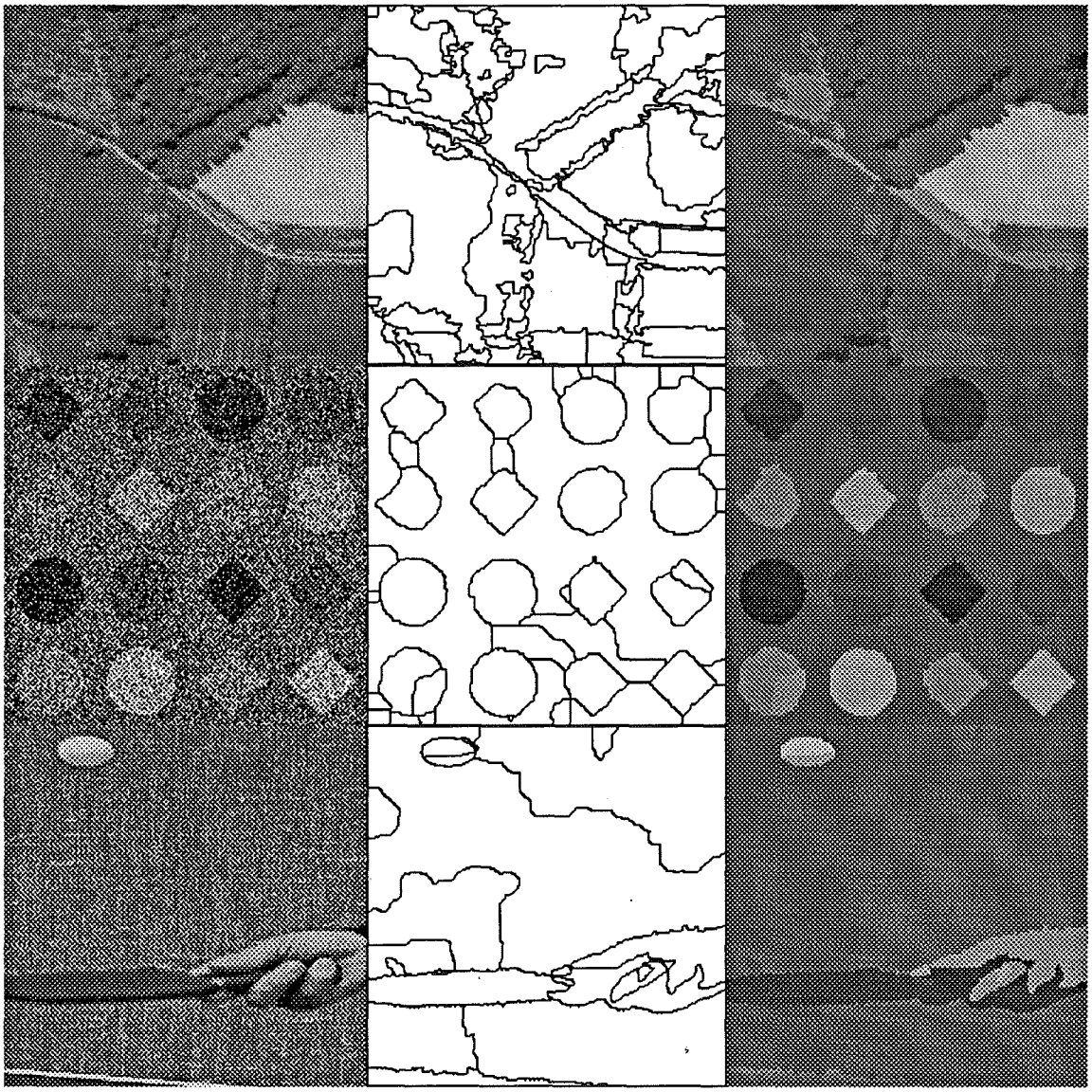

Fig. IV.12.- Examples of segmentations of textured images

The first example of Figure IV.12 deals with the segmentation of the image Aerial (T = 0.8, 108 regions). Actually, this image presents some areas that are clearly textures (e. g.: the forest) while others could be taken as non-textured ones (e. g.: the crops). Although textured areas are not detected as single regions, they are split into only few regions. However, a problem arises with some non-textured areas since they are also split or not detected. The second row shows the segmentation of a synthetic image (T* = 0.6, 43 regions), which will be refereed as Synthetic. This image is formed by a constant gray

level background and 16 regular figures (8 rhombuses and 8 circles) with different constant gray level values plus a high variance uniform noise (σ = 50). Note that, in spite of having detected more than the 17 actual regions, the segmentation has succeed in finding the basic structure of the image, even with such a strong noise. Finally, a subimage of a frame of the so-called Table Tennis sequence has been segmented (T* = 1, 22 regions). In this case, the segmentation of the textured area is achieved by splitting the area into few regions. On the contrary, details on the hand, the table and the racket are overlooked.

IV.3.5.- Setting manually the value of T*

In order to study the possible improvements of using different T* values at each decomposition level, a supervised segmentation task is carried out. That is, starting for the top level of the pyramid, several segmentations are performed at each level by varying the T* value. The best segmentation is selected, following the quality criteria discussed in the previous section. This result is used as initial segmentation for the following finer level. The procedure is repeated down to the original image is segmented. A diagram illustrating this procedure is shown in Figure IV.13 .

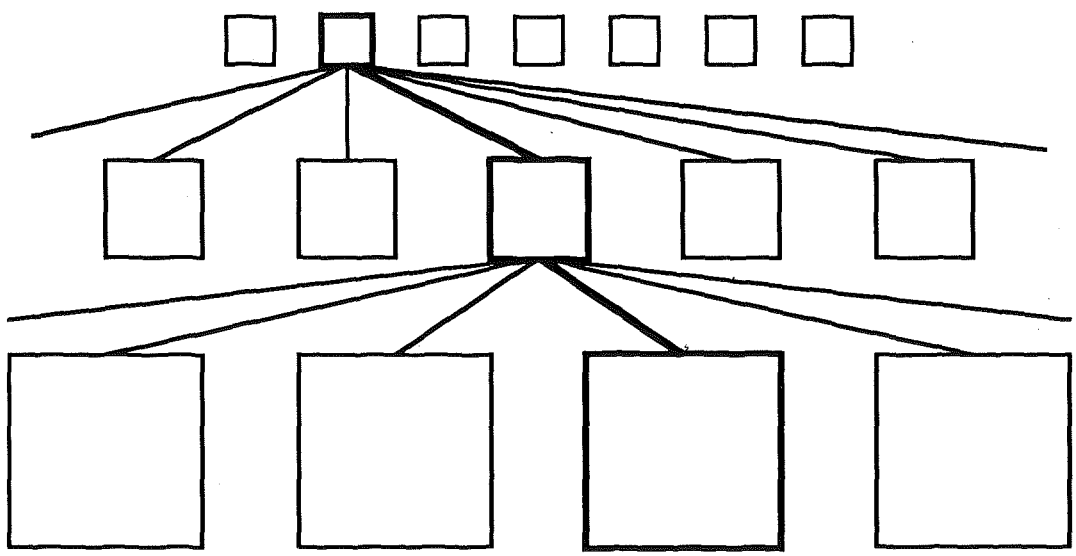

Fig. IV.13.- Supervised procedure for choosing the best values of parameter T*

It has to be noticed that an exhaustive study of all the combinations of possible values of T*(l) is rather prohibitive. In addition, the choice of the best segmentation at each level of the pyramid is a little reliable task, specially at coarse levels of the pyramid. The complexity of this selection is illustrated in

Figure IV.14, where this procedure is shown for levels four, three, two and one of the Gaussian pyramid of the Cameraman image. These results are presented as mosaic images so that the difficulty of the subjective selection is highlighted. Note that differences between two consecutive segmentations at the same level are almost unnoticeable. Therefore, only general ideas about the behaviour of the segmentation results when varying the T* value will come from this supervised study.

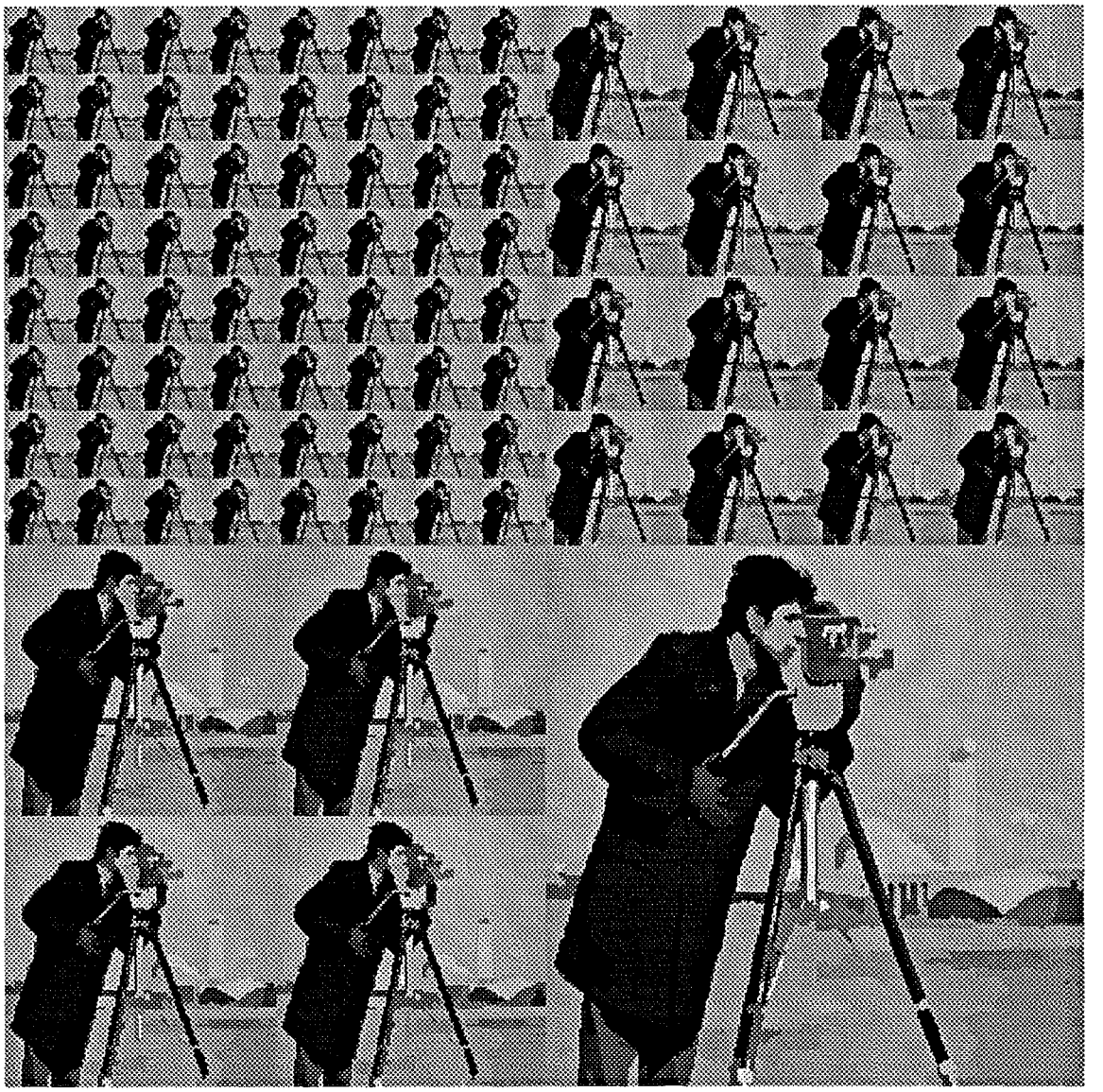

Fig. IV.14.- Supervised selection of T*

The first point that arises from this study is that nearly the same segmentations are yielded by using any value of T*(4); that is, when segmenting the top of the pyramid. Actually, the same statement has been

made in Section IV.3.1 when choosing the initial segmentation for the coarsest level procedure. Another important point is that, in all cases, the chosen value for T*(0) is close to one. This result confirms those obtained by the study performed in Section III.3.2 about the influence of the temperature parameter in the monolevel case. Indeed, both studies are analogous since they deal with the obtaining of the final result, given an initial segmentation.

A new result is obtained regarding the different tendency of the T*(l) parameter through the decomposition when handling textured and non-textured images. Mainly, the chosen temperature parameters increase their values when going down through the pyramid in the case of textured images and vice versa for non-textured images. In the case of images that cannot be classified clearly into one of these groups because they contained both kind of regions, the variation of T*(l) follows that of non-textured images but with a less marked tendency. This result is illustrated in Table IV.2 where the T*(l) values chosen for six different images are shown (two containing very few textured regions, two containing textured and non-textured regions in a similar amount and two containing mainly textured regions)

	Miss Am.	Building	Cameram.	Aerial	Table-Ten.	Synthetic
T(4)	5	2.5	1.4	1.2	0.4	0.3
T(3)	3	1.7	1.2	1.1	0.5	0.3
T(2)	2	1.2	1.1	1.1	0.6	0.5
T(1)	1.7	1.1	1	0.8	0.7	0.5
T(0)	1.2	1	0.9	0.8	0.8	0.7

Table IV.2.- Different behaviour of T*(l) in the case of textured and non-textured images

The decreasing tendency of T* when dealing with non-textured images can be understood paying attention to the fact that the higher the level in the pyramid, the stronger the filtering of the image. In this way, assuming non-textured images (that is, images with homogeneous gray level regions), the filtering procedure almost does not change the data within each region while it does blur contours between regions. Therefore, at coarsest levels of the pyramid, the image model should give priority to interior information with respect to boundary information. This priority is given by decreasing the value of the temperature parameter T* as the level index (l) decreases.

In the case of textured regions, the filtering has an additional effect. Textured areas are more locally homogeneous after filtering and their