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 I

INTRODUCTION

There exist many different definitions of image segmentation in the literature. Actually, almost each author has coined one. The definition used throughout this work is the following: segmentation is a process that divides completely an image into a set of homogeneous, connected regions related to the objects in the scene. Therefore, the result of a segmentation is a labelled image, corresponding each label to a different region. Note that the above definition is rather vague since it fixes neither the criterion of homogeneity to be used nor the algorithm to be applied. However, there are two clear constraints which have been introduced: the completeness of the segmentation (to ensure the total representation of images) and the connectivity of the regions (to differentiate the segmentation problem from the classification one).

Several segmentations can be found for a given image satisfying the above definition. Therefore, some criteria have to be proposed in order to estimate the segmentation quality. In [1], the authors have established a set of rules giving, rather than a criterion, a guide-line for a good image segmentation. These are: "Regions of an image segmentation should be uniform and homogeneous with respect to some characteristic such as gray tone or texture. Region interiors should be simple and without many small holes. Adjacent regions of a segmentation should have significantly different values with respect to the characteristic on which they are uniform. Boundaries of each segment should be simple, not ragged, and must be spatially accurate." As it can be seen, this set of rules is rather qualitative and as little concrete as the given definition.

I.1.- Importance of segmentation: its applications

The vagueness of the definition is, of course, on purpose. The range of frameworks in which segmentation is nowadays utilised is so wide that a more concrete definition would not apply for all of them. Indeed, almost each discipline dealing with images uses some kind of segmentation as low level processing. Moreover, the success of the segmentation algorithm often determines the success or failure of the overall image analysis algorithm. As examples of applications, the following non-exhaustive list may be given:

- Image enhancement: for images processed by human viewers as in cytology for detecting cells, in traumatology for the study of X-ray images, localisation of fractures or in analysis of echographies and CAT's; in order to aid machine performance as in industry for carrying out industrial processes automatically, such as control of quality, surveillance or default detection.
- Image restoration: for obtaining zones in the images with similar characteristics which have to be restored in a similar way.
- Image coding: for studying the motion within a video sequence by following regions and computing their motion vector field; for detecting homogeneous regions within the scene and code each one separately.
- Image understanding: in environment studies for classifying the different kinds of vegetation within an area and, by means of satellite frames, to study their possible degradation; in robotics for detecting objects in a scene, in order to create a tridimensional model and for leading the movements of a robot within this scene; in agriculture for classifying and for controlling crops within a zone by means of images taken from a satellite or a plane.

This large set of applications provides with a huge number of possible kind of images to be segmented, with very different characteristics from each other. In this way, segmentation has to handled from noiseless, binary images up to textured, multiespectral ones, going through images with illumination problems or very noisy ones. All these possibilities not only lead to the necessity of a vague definition, but also make the task of segmentation very difficult. That is, to devise a segmentation method performing well when applyed to the whole aforementioned set of applications seems rather impossible. On the other hand, to develop a tailored method for each segmentation problem can be afforded. Actually, what one would like to have is a general-purpose method that, performing fairly well in each single problem, could be easily tailored to each one.

I.2.- Intrinsic problems of segmentation

The large range of possible images is not the only fact that makes segmentation a difficult task. Some problems arise when validating a segmentation or comparing it with other results. The first problem comes from the fact that two different projections are carried out, when acquiring a picture. Physical world, which is a three dimensional space, is mapped into an image, which is a two dimensional space. Furthermore, each point in the scene has associated a three dimensional vector representing its colour. This colour space is, on its turn, projected into a one dimensional space, in the case of black and white images. Thus, the act of acquiring a picture F[·] involves two different projections which produce the loss of some information:

$$\begin{array}{lcl} \textit{real world} & \xrightarrow{F[\cdot]} & \textit{image} \\ \Re^3 \text{ (scene)} & \longrightarrow & \Re^2 \text{ (picture)} \\ \Re^3 \text{ (colour)} & \longrightarrow & \Re \text{ (black and white).} \end{array} \tag{I.1}$$

Therefore, segmentation can be seen as an ill-posed problem [2]. That is, given a scene and acquired an image from it, the segmentation of such image may not be unique and, even more, once a solution has been chosen, it may not represent the objects in the original scene. Furthermore, if a rather complicate image is given to a set of individuals and they are asked to segment it manually, very likely, one may get as many segmentations as individuals are in the set. Therefore, the validation problem leads to the use of either synthetic images from which the segmentation result is known exactly (see Figure I.1), or a set of trained people to rely on.

This second solution links with another problem of segmentation: the comparison of an artificial segmentation with the segmentation performed by a human observer. Human visual system is, by far, the best known tool for segmenting 2D spaces [3]. This skill makes very difficult for the observer to understand why a machine is not able to achieve the same kind of segmentation that she/he is able to. People have a tendency to underestimate the complexity of segmentation and that leads to a clear disappointment when checking the results obtained by any artificial computation.

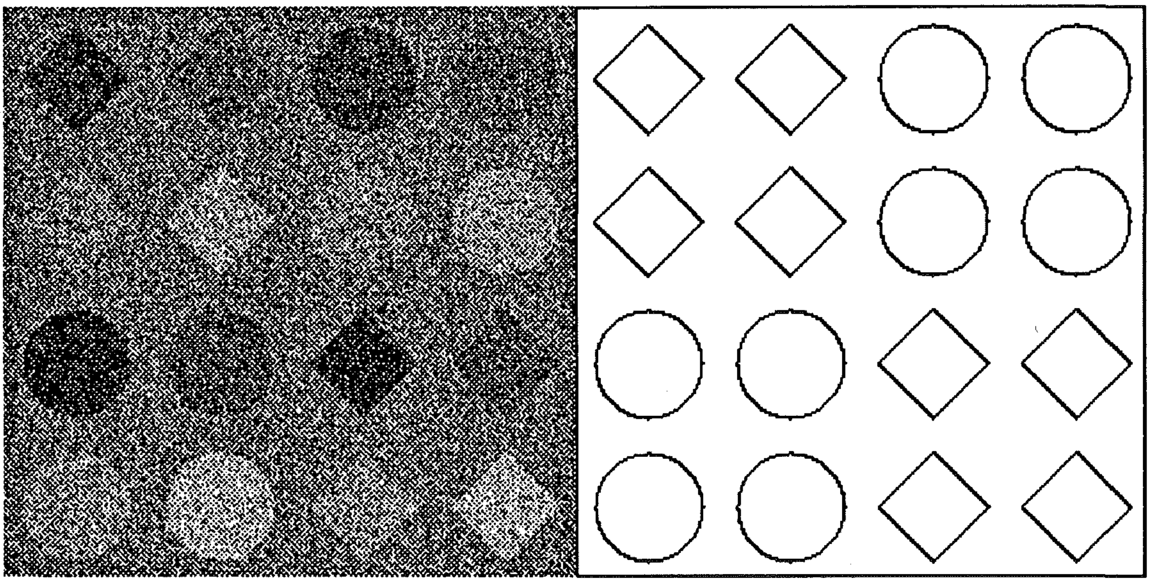

Fig. I.1.- A synthetic image and its exact segmentation

A possible solution for solving this machine handicap is to try to emulate the human visual system. There has been a large number of studies [3, 4, 5] in this field, but very little has been concluded due to the difficulty of the problem. Most of these studies has handled the problem from a qualitative or philosophic viewpoint, without giving a quantitative model or theory for characterising the human visual system. Furthermore, when some theories have been built for a specific environment, their application to more general or realistic situations have failed. Hence, their results do not help to implement computer vision algorithms.

Nowadays it is broadly accepted that vision involves not only low level procedures (such as contour detection, segmentation or texture classification) but also makes use of high level knowledge. These two levels of processing work in parallel, exchanging information and producing two-directional feedbacks in such a way that the comprehension of the image is achieved. This idea comes from the application of Gestalt theories to the framework of human visual perception [6], and it allows understanding how human beings are able to see objects in a given scene whose symbols are not explicitly present.

Figure I.2 shows a square built with a set of circles and, in its interior, a circle consisting in a set of squares. The human ability for detecting such high level structures cannot be easily imitated by an artificial system. Even more, in some applications one may want the computer to detect 24 circles and 8 squares in the scene (low level structures), rather than a single square and a single circle (high level structures). Hence, two ideas have to be highlighted: the simulation of the human visual system for segmenting is a tough task (actually, not solved) and it may not be the eventual goal of an image segmentation algorithm.

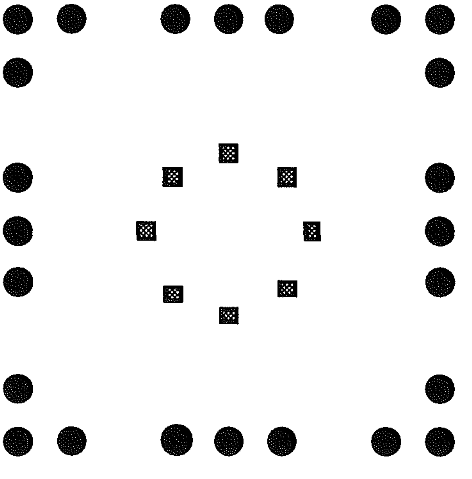

Fig. I.2.- An image with a square and a circle not explicitly present

The last intrinsic problem, but not the least, is the uncertainty present in the images [7]. This uncertainty may come from different sources. For instance, in the process of sampling and quantisation of images, some uncertainty is introduced. These processes yield elements in the image that do not belong clearly to any region; that is, these elements correspond to areas of transition between homogeneous zones. An analogous effect can be produced by the addition of noise to the image. In this case, noisy pixels do not belong to a transition zone, but their characteristics do not match with those of the surrounding regions. In either case, the problem of labelling these pixels has to be solved.

I.3.- Formal definition of segmentation

In spite of the vagueness of the previous given definition of segmentation, a basic formulation can be given. This formulation consists of the following five conditions that a segmentation has to fulfil:

$$
\begin{aligned}
&\text{A)}\quad I = \bigcup_j R_j \\
&\text{B)}\quad R_j \cap R_i = \varnothing \quad \forall\, j \neq i \\
&\text{C)}\quad R_j \quad \text{connected} \quad \forall\, j \\
&\text{D)}\quad P(R_j) = \text{True} \quad \forall\, j \\
&\text{E)}\quad P(R_j \cup R_i) = \text{False} \quad \forall\, j \neq i
\end{aligned}
\qquad (I.2)
$$

where I represents the whole image, $\{R_j\}$ the set of regions and $P(\cdot)$ a homogeneity predicate.

Condition A) requires the segmentation to be complete; that is, each pixel in the image is labelled as belonging, at least, to one region. On its turn, condition B) demands each pixel to be assigned, at most, to one region. Therefore, every pixel must belong to one and only one region (each pixel must have one and only one label). The constrain of connectivity within a region is introduced in condition C). These first three conditions define a partition of an image. The two remaining conditions control the homogeneity of regions: condition D) introduces the predicate $P(\cdot)$ testing whether every region in the segmentation is homogeneous, while condition E) ensures that any merging of two existing regions does not form a new region (it will not comply the predicate). It should be noticed that the vagueness of the definition still remains since the homogeneity predicate has not been fixed.

Some authors [8, 9] introduce this formulation of the segmentation problem just for some algorithms of segmentation, making a distinction between "complete" and "incomplete" segmentations. It can be said that an "incomplete" one is not really a segmentation, since it does not fulfil some very important requirements that a segmentation should (in some cases completeness, in others connectivity of the regions). Moreover, note that the above five conditions are almost a translation into a formal language of the definition of segmentation given at the beginning of this chapter.

I.4.- Segmentation approaches

Once the segmentation problem has been stated, a non-exhaustive overview of the existing approaches will be done. Segmentation approaches are usually classified into two main groups: edge-based techniques and region-based ones. As their names indicate, the first group looks for edges within the image (areas representing discontinuities of the homogeneity criterion), while the second group seeks regions (zones accomplishing the homogeneity criterion).

I.4.1.- Edge-based techniques

This kind of techniques look for discontinuities of the homogeneity criterion. Since almost all of them are based on gradient estimation, the assumed criterion relies on gray level homogeneity of regions. These techniques consist usually of four different steps. Figure I.3 shows a block diagram of these four steps.

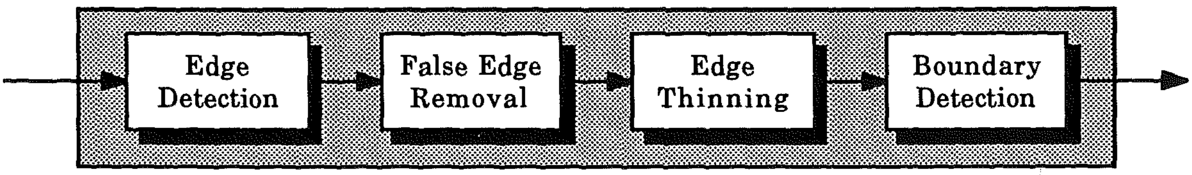

Fig. I.3.- Block diagram of an edge-based segmentation

The first step is the edge detection. This detection can be achieved by several algorithms, which can be classified into four categories [10]:

• Gradient operators: basically, these algorithms use a discrete approximation of the gradient or of the laplacian operators. The approximation of the gradient in each direction (directional difference) is performed by means of the convolution of a small kernel with the image. A scalar measure of the local gradient can be estimated in different ways: taking the maximum, the absolute value or the magnitude of the gradient. Changing the size of the kernels, theirs weights and the estimate, one can obtain different gradient operators: Roberts [11], Sobel [12], Prewitt [13], etc... Another possible approach is to substitute linear convolutions for morphological operators [14, 15]. All these algorithms have the problem of being very sensitive to noise. In order to solve this problem a new theory of edge detection has been proposed [16], which leads to methods combining image filtering and laplacian operators. These methods yield more robust estimations of the gradient and therefore, less noisy edges.

• Template matching: in this approach, a set of kernels is used in parallel (selecting the output of one filter for each pixel) rather than by combination (obtaining a result by combining all the outputs, as in the previous approach). Thus, a set of templates, each one representing a kind of edge (different directions), is applied on every point of the image. At each point, the template producing the highest correlation gives the edge characteristics at this point. Several sets of templates have been proposed, for instance the so-called Kirsh template [17] or the Frei&Chen template [18].

• Edge fitting: the basis of this approach is the assumption that an edge can be modelled as a step discontinuity. Using this model, regions in the image whose edge fitting error is small are sought. The parameters of the model which better fits this region determine the position, orientation and magnitude of the edge. Two main techniques have been developed following this idea, namely Hueckel [19] and Nalwa and Binford [20]. It has to be pointed out that these techniques require more computation than the above presented techniques while not improving their performance.

• Statistical techniques: here, the detection of an edge is treated as a hypothesis testing. A line going through a window is said to be an edge if the two sets of pixels that it forms on its sides can be defined as two different regions. The study has to be carried out for different orientations in the same window, and the orientation which best accomplishes the two regions hypothesis is chosen. Yakimovsky [21] presented a method using a 7x7 window and normal models for regions.

It has to be said that the different methods above presented lead to similar results. As an example of gradient detection, Figure I.4 shows the morphological gradient of an image.

Once the edge detection has been performed, the second step in the segmentation process deals with removing false edges. The necessity of this step is twofold. Firstly, since edge detectors are very sensitive to noise, small spots appear in edge images. This kind of artifacts can be seen in Figure I.4: on the bottom righ-hand corner the grass has yielded some false edges. Hence, the need of an algorithm for removing small, isolated spots. Secondly, almost all the aforementioned techniques yield a gray level image. That is, in edge images the estimated magnitude of the gradient is represented. Therefore, a threshold must be introduced for removing weak gradient points and for obtaining a binary edge image. In Figure I.4 some weak gradients can be seen within the coat or the pants of the man.

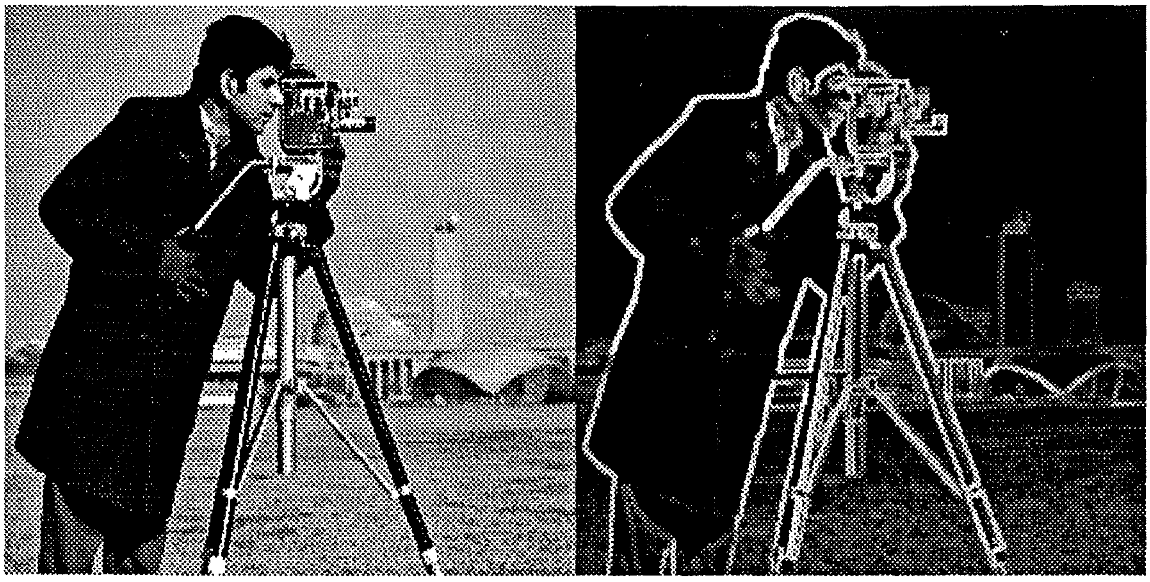

Fig. I.4.- The original Cameraman image and its morphological gradient

The third step of an edge-oriented segmentation tries to solve the problem of having wide edges. In order to obtain a segmentation, contours must be represented by one-pixel width edges. Nevertheless, the output of the edge detectors is not constrained to have such width and, due to the image uncertainty, algorithms yield wide edges (this effect can also be observed in Figure I.4). Thus, a thinning step is mandatory. In segmentation applications, thinning techniques present some problems, e.g.: usually, they do not take into account the original image but just edge images; after thinning, the connectivity of edges may change. Thus, conventional thinning algorithms cannot be used. A review of more suitable thinning techniques for this application can be found in [22].

The last step of an edge-oriented segmentation creates a partition of the image from its edge map. This procedure is usually refereed as gap filling, edge linking or boundary detection. It can be carried out just taking into account the information contained in the edge map or combining this information with that of the original image. Anyway, it has to be noticed that whereas the first three steps are mainly filterings and therefore simple, low-level operations, this last step requires the use of some heuristics or assumptions on the kind of boundaries that are wanted. Among these techniques there exist different classes: boundary refining, the Hough transform, graph searching, dynamic programming and contour following . A more extensive review of such techniques can be found in [9, 10, 23].

The above four steps are grouped in only two by some authors [9, 12]. That is done by taking edge detection, thresholding and thinning as a single procedure. In this way, edge-oriented segmentations are split into a low-level and a high-level stage. The more sophisticate the edge detection, the more simple the boundary detection can be. Edge-oriented segmentations are not simple to use (note that each step in the process presents problems, as well as their concatenation) and, moreover, they do not achieve high performance.

I.4.2.- Region-based techniques: Thresholding

In the segmentation definition given in the introduction of this chapter, there are two main constraints that a region should fulfil: connectivity and homogeneity. A segmentation obtained via image thresholding exploits the second constraint by using the most simple concept of homogeneity: gray level uniformity. In this way, thresholding is an operation only accounting for the gray level value of pixels $g(x, y)$, without involving their position within the image. Actually, when a thresholding $T(\cdot)$ uses only the gray level information of a point in order to obtain the different thresholds, it is called a global thresholding: $T(\cdot) = T(g(x, y))$. This is the simplest kind of thresholding which can be used, and this simplicity leads to a very fast implementation using Look Up Tables (LUTs):

$$T(g(x, y)) = L_i \qquad \text{for} \qquad Th_{i-1} \leq g(x, y) < Th_i \ , \tag{I.3}$$

where $\{L_i\}$ represents the set of labels and $\{Th_i\}$ the set of thresholds.

Note that the segmentations obtained by thresholding fulfil conditions given in (I.2). In particular, the conditions of completeness (A), validity of current segmentation (D) and no possibility of further merging (E) are accomplished by construction. In order to guarantee the uniqueness condition (B), the symbols "greater than" and "equal or greater than" in (I.3) have to be strictly imposed. On its turn, condition (C), which copes with the concept of connectivity, has to be imposed once the thresholding has been applied. That is, a set of points is said to be a region if, and only if, after thresholding, all of them have the same label and are connected.

The major problem with this technique is that of selecting the set of thresholds $\{Th_i\}$ to be used. General inspection of the histogram seeking the right location of thresholds assumes that the histogram is clearly multi-modal, which is seldom true. Rather, histograms are almost uni-modal (see the discussion of the role of illumination in [8]) or noisy enough to hide their multi-modal structure.

Some techniques have been proposed to solve this problem. For noisy multi-modal histograms, the application of a simple smoothing filter may help to show up the histogram real structure. However, this technique does not lead to any improvement when handling uni-modal histograms. Another possibility is to use a local thresholding. Here, the thresholds no longer depend only on the gray level of the pixels, but they account also for some neighbourhood properties $N(x, y)$. That is, $T(\cdot) = T(N(x, y), g(x, y))$. For instance, the histogram of pixels laying on or near a boundary in the original image can be computed. In this way, the influence of a region in the histogram depends less on its size. Furthermore, the gray level values of these pixels are typically between the gray level values of the regions that they separate and, therefore, they are good estimates of the set of thresholds. Of course, this technique requires knowing where the regions are. In order to shortcut this problem, an estimate of the gradient of the image is computed; that is, some local information $(N(x, y))$ is utilised in the process.

A third kind of thresholding is the so-called dynamic thresholding. In this case, the set of thresholds can change from one point to another within the image. Therefore, the thresholding depends on the coordinates of the points as well as on the other features used before: $T(\cdot) = T(x, y, N(x, y), g(x, y))$. This variant has been introduced to allow a more local image analysis. The analysis is implemented by dividing the image into a set of subimages (usually overlapping), and by studying each one separately. In this way, a small, bi-modal zone of the image can be detected since the rest of the image does not interfere in its study.

As it has been stated previously, segmentation via thresholding only makes use of the concept of homogeneity of a region (actually, the simplest kind of homogeneity). It has to be noticed that the thresholding process is more related to a classification than to a segmentation. The histogram of an image is divided in zones so that a classification of the pixels within the image is performed. Once an image has been thresholded, the segmentation is obtained by grouping the neighbour points with same label. Therefore, the concept of connectivity of a region is not exploited but only introduced a posteriori. One might think that, rather than a region-oriented method, thresholding is a point-oriented method, since the concept of region is poorly used. In fact, the improvements over the basic thresholding above presented try to overcome this problem introducing some spatial information in the procedure.

Thus, the drawbacks of segmenting by thresholding come from its main advantage: its simplicity. Simplicity in the kind of homogeneity that it assumes (a usual scene does have more complicated regions than quasi gray level constant ones) and in the use that it makes of the neighbourhood (the neighbours of a given pixel can provide with much more useful information for segmenting that just their location). Therefore, logical improvements to introduce are the definition of more complicated homogeneity criteria and a better use of the neighbourhood information.

I.4.3.- Region-based techniques: Region growing

A method taking into consideration the concept of connectivity while performing the segmentation is the so-called region growing [28]. The method starts by assuming a large set of initial, small regions (usually, each pixel in the image is a single region). The segmentation is then performed by the successive merging of small, neighbour regions sharing the same kind of homogeneity. Each merging produces a larger region still satisfying the homogeneity predicate. This procedure is followed until no new merging is allowed. Therefore, it is a bottom-up method, since progress is made from small regions (or even isolated pixels) up to large regions.

On the one hand, the conditions of completeness (A), uniqueness (B) and connectivity (C) in (I.2) are directly satisfied by construction, so that, throughout the whole segmentation procedure, a partition of the image is available. Moreover, since in the initial situation the partition is formed by very small regions or even single pixels, condition D) is almost ensured (very poor should be the chosen predicate in order not to work in such a simple situation). On the other hand, condition E) is not initially fulfilled. Actually, the final segmentation is achieved by checking this condition for each possible merging.

The main advantage of this method is that it exploits both concepts at the same time: homogeneity and connectivity. It has to be noticed that here, unlike in the thresholding case, the homogeneity criterion is not fixed. Thus, this method provides with much more flexibility. This flexibility has opened the way to a large set of different techniques ranging from simple clustering to stochastic boundary estimation [24]. However, none of these techniques has turned out to be a general purpose one.

Region growing techniques present some inherent drawbacks. The main one is that, like many other nonlinear optimisation problems (segmentation can be faced from this viewpoint), it is very depending on the initial conditions; that is, the starting set of regions. In fact, not only the initial conditions can bias very much the final segmentation, but also the order in which the mergings are performed.

This problem is better explained with the help of Figure I.5, where two possible segmentations of the same image are shown. The segmentation predicate is defined as "a set of connected pixels $\{p_i\}$ defines a region if the mean absolute error between the pixels and their mean gray level value is lower than a given value M:

$$\{p_i\} \text{ is a region if } \{p_i\} \text{ connected and } M > \frac{1}{C[\{p_i\}]} \sum_{x \in p_i} |m_x - x| \text{ ",}$$

where C[·] stands for the cardinality of the set and m_x is the mean gray value of the set of pixels.

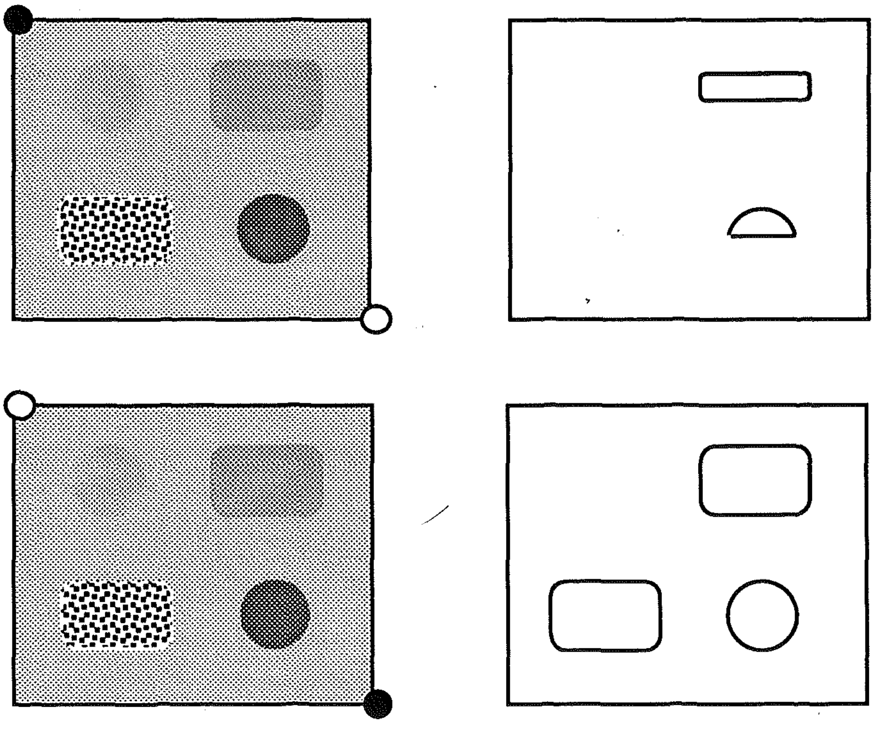

Fig. I.5.- Effect of the sorting when merging on a region growing procedure

The mergings in Figure I.5 have been carried out following the sorting given by a scanning starting in the black spot and ending in the white one. As it can be seen, in both cases the region corresponding to the background overgrows taking pixels belonging to other regions whose gray level are close to this of the background. Beside this effect, some regions with gray values not so close are wrongly segmented in the first case. This wrong segmentation is owing to that each time a point belonging to another region is introduced in the background its mean value increases. In the second case, by just varying the sense of the scanning, this second effect is avoided.

In order to solve these problems, one possibility is to fix some starting points within the image, and to perform a merging for each point following a given order. In this way, if a starting point is fixed for each region (this is the concept of seed or marker of a region), the sorting for the mergings would ensure that each region grows fairly, and no overgrowings will appear. Nevertheless, this method assumes that the region locations are known; that is, the segmentation is known a priori. A usual practice is to give a set of points randomly distributed within the image as seeds.

A way to fix a sorting in the merging process, overcoming the necessity of providing with a set of seeds, is to associate to each merging a cost function. Given a stage in the merging procedure, for each possible merging to be performed, a cost of this merging is computed. At each step, the merge with the lowest cost will be carried out. This technique ensures that no overgrowing is produced, but it has the problem of computing the costs, which is very time consuming. Even more if one considers that, after each merging, costs must be updated.

Several cost definitions have been proposed. Usually, the cost criterion is related to the definition of the homogeneity predicate. The cost of a merging is greater when the resultant region is closer to fail the predicate. Following this idea several possibilities have been proposed; for instance, predicates relying on gaussianity of regions [25] or based on modelling of regions by means of polynomial surfaces [26]. In these cases, merging costs are defined by computing how far the resultant region is from fulfilling the assumed model. For a more extensive discussion on this topic, see [25].

Another inherent problem of region growing is that it uses a complicated data structure. Note that at each moment, the data structure has to keep, for each region, records of its features, as well as a list of its neighbourhood regions, in order to perform the mergings. This kind of data structures is usually refereed as Region Adjacency Graphs (RAG) [27]. In the improvement of the method presented before, this data structure is even bigger, since it has to contain also information about the costs of mergings.

Finally, a third problem comes from the fact that region growing is a bottom-up technique. Due to this characteristic, the view that algorithms have of images is very local and hence, the use of features involving large areas is difficult to apply. This difficulty leads to oversegmenting textured areas since textured regions cannot be characterised from a local point of view.

It is worth noting that, for industrial applications, region growing techniques are considered too computational time demanding and, therefore, too slow to be implemented, in spite of being known that they can achieve good results.

I.4.4.- Region-based techniques: Split and Merge

In the previous section the importance of using, at the same time, the homogeneity criterion and the concept of connectivity has been highlighted. However, it has also been shown that when making use of a bottom-up technique, such as region growing, some problems arise. Trying to solve these problems, a method, still dealing with homogeneity and connectivity together but combining top-down and bottom-up approaches, has been proposed. This method is the so-called "split and merge" approach [28].

This approach is initiated by taking, unlike in the region growing, the whole image as a single region. The homogeneity predicate is tested on this initial region and, if it fails, the region is split into four equal, disjoint regions. If the test does not fail, the region is accepted as a right one. Assuming that the original image is square (and usually of dimensions power of 2), the splitting is performed by dividing the initial region into four squares. Each of these new regions have as side half the side of the initial one (division into quadrants). Other ways for subdividing the image are presented in [25]. The algorithm is iterated until all regions fulfil the homogeneity predicate or until a prefixed size of region is reached. At this moment, the splitting stage is finished. Since the algorithm starts analysing large regions and progressively goes to smaller ones, this stage is said to be a top-down process.

Note that this splitting step clearly relies on the quadtree data structure [27]. In this tree, the root represents the entire image and the leaves represent the different quadrants. In Figure I.6, the splitting of an image into homogeneous regions can be seen as well as the corresponding quadtree (example taken from [29]). In this tree, squares denote regions that do satisfy the homogeneity criterion, while circles denote regions that do not (and therefore, should be further split).

As in the previous case, the conditions of completeness (A), uniqueness (B) and connectivity (C) of (I.3) are satisfied by construction. Therefore, at each stage of the procedure (that is, at each level of the quadtree), a partition of the image is obtained. On the other hand, conditions (D) and (E) are not accomplished. In fact, the splitting is performed by verifying the current segmentation -condition (D)-, whereas the merging validation (E) is not carried out at this stage of the segmentation.

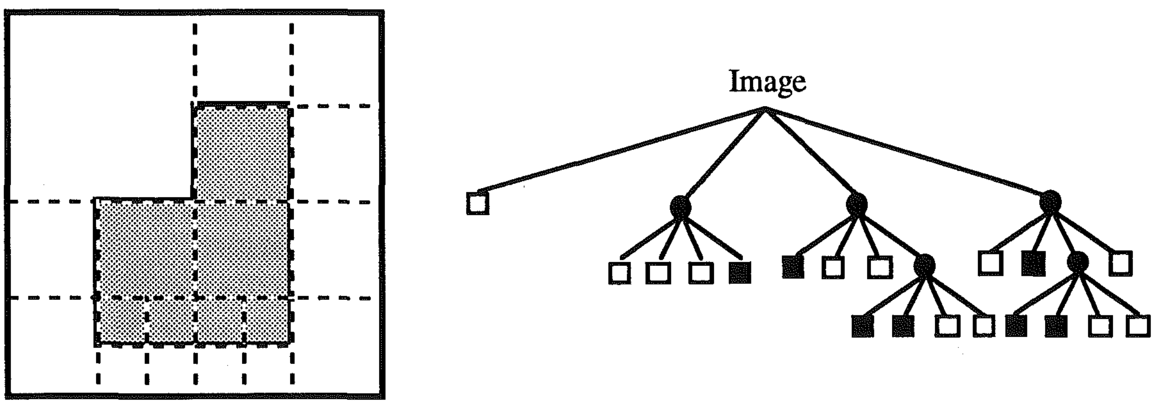

Fig. I.6.- The splitting procedure of a very simple image and the corresponding quad-tree

Once the splitting has been carried out, a merging step has to be performed. The necessity of this stage comes from the fact that condition (E) has to be satisfied. Adjacent regions that have been detected separately as being homogeneous can actually be parts of a bigger region. This can be clearly seen in Figure I.6 where all the homogeneous regions (marked as a block square in the quadtree decomposition) compound only two regions: the figure and the background. In order to execute the merging stage, a new data structure has to be built. This need is owing to the lack of information about adjacency among regions in quadtrees. Hence, a RAG, as discussed in Section I.4.3, must be used. As in region growing techniques, condition (E) tests whether two regions should be merged or not. Several variations on this basic method have been proposed; for instance, a hybrid technique in which splitting and merging alternate at each level of the quadtree [30].

One may consider "split and merge" techniques as the first trial to introduce multiresolution analysis in segmentation. Though it is true that images are analysed at different resolutions, since the analysis is very constrained (it is performed separately at each quadrant) its possibilities are not fully exploited. However, "split and merge" yields some improvements which have made it very popular (it has to be said that it is the most widely used segmentation technique). One of its advantages is that it is a concatenation of a top-down and a bottom-up process. Thus, it permits to use jointly a global (in the splitting step) and a local (in the merging one) view of the image. Furthermore, it reduces the dependency on the initial conditions, given that the initial stage is based on a top-down approach. Finally, as in the region growing methods, it allows using any kind of homogeneity predicate: from very simple homogeneity criteria [31] up to model-guided ones [32]. Thus, the "split and merge" algorithm represents a big step towards the ideal algorithm described at the end of Section I.1.

Spite the above advantages, this algorithm does not solve all the segmentation problems, on the contrary it presents some relevant drawbacks. Since at each level a region not satisfying the homogeneity predicate is split into its quadrants, the algorithm has a tendency to form very rectangular-like regions. The merging process alleviates this "block" effect but it does not eliminate it. Furthermore, as it has been commented, this characteristic leads to an incomplete exploitation of the multiresolution analysis.

In addition to that, each stage of the algorithm (the splitting and the merging) requires a different data structure. Therefore, the implementation of the algorithm is not straightforward. However, although the data structure complexity in these algorithms is comparable, and even bigger than that of region growing algorithms, its volume is smaller. The reason of this reduction is twofold: first, the quadtree compacts the initial information and, second, the starting number of regions in the RAG is sensitively lower. The refinement on the basic "split and merge" before commented [30] can be used in order to improve the performance in this aspect. By limiting the mergings to regions that, being at the same level of the tree, are descendants of the same father, the same quadtree structure can be preserved until the final merging step. Nevertheless, the RAG data structure is still required for this last step.

It must be highlighted that the best results, using either region growing or "split and merge" techniques, are achieved when the algorithm makes use of a fairly complex model-based criterion of homogeneity [33] (best results meaning not only best performance on a given picture, but also referring to performance when using it as a general purpose algorithm). In this way, the algorithm can take into account simultaneously several features describing regions. However, this approach is, of course, very time consuming and, moreover, when using model-based criterion and "split and merge" techniques, one has to be very careful defining the stopping criterion (starting situation for the region growing case). The use of a overly restrictive predicate can lead to a final segmentation where each single pixel represents a region.

I.4.5.- Region-based techniques: Linked pyramid segmentation

Another approach for segmenting images is to face the problem from the uncertainty point of view [7]. Here, unlike in the previous techniques, the tasks of segmenting an image and of estimating characteristics of its regions are not stated separately. On the contrary, they are assumed to be highly interdependent and, therefore, they are performed jointly. Obviously, given that both problems cannot be solved together in a single step, iterative algorithms have to be proposed.

As it has been shown in Section I.4.4, multiresolution analyses may help to improve the segmentation performance: characteristics of regions defined at different levels of resolution can be studied simultaneously. Furthermore, by using such kind of analyses, the uncertainty effects can be minimised [7]. Multiresolution approaches allow the study of images at different levels of detail. In this way, the uncertainty present at the lowest levels is usually removed at the highest ones.

Taking into account the above aspects, as well as the drawbacks of a quadtree structure, the linked pyramid data structure has been defined [34]. This data structure is a layered arrangement of successive smaller replicas of the original image, which is at the bottom of the structure. Usually, the reduction from one level to another is by a factor of two. Unlike in the quadtree, a datum in a level (k) is linked with more than one datum at level (k+1); that is, in tree terminology, a son may have, at least initially, more than one father (candidate fathers). Therefore, two elements at level (k+1) can have regions of support overlapping at level (k). This multiple connection among elements is what helps to cope with the uncertainty of the segmentation: a pixel is not biassed to belong to a unique region, as it happens with the quadtree structure. Moreover, due to this multiple connection, there is no restriction to the shape of regions obtained by this algorithm, avoiding in this way the "block" effect produced by "split and merge" algorithms.

Given a linked pyramid data structure, the segmentation of an image is performed by computing the properties of each node based on the average of its sixteen children through all the levels in the pyramid. Note that an element at level (k) contributes in the properties of four candidate fathers at level (k+1) and that these computations are made in a bottom-up fashion. Once the links have been calculated, they are redefined by taking into account the similarities between children and candidate fathers. That is, if a child has characteristics differing very much from these of one of its candidate fathers, the link between them is either removed or weakened. Afterwards, properties of each element are redefined relying on the new links. These two steps are iterated until no new change is introduced in the data structure. The convergence of the algorithm is guarantied, since it can be seen as a special case of the ISODATA algorithm [35]. Actually, the whole technique could be formulated in terms of Fuzzy Set Theory [36], assuming that links between children and fathers are a special case of membership functions [37].

In relation with the formulation of (I.2), it is worth noting that in this case the uniqueness condition (B) is not satisfied throughout the process. That is, since a pixel may be related to more than one candidate father, the intersection between two regions is not the empty set. Therefore, the method provides with a partition only at the end of the procedure. Furthermore, the algorithm above presented does not ensure the connectivity of regions. In Figure I.7, a possible result of applying this algorithm to a monodimensional signal is shown. In this result, it can be observed that two unconnected regions are produced (regions represented by black and white circles). This problem is owing to the above commented flexibility in the shape of regions produced by this algorithm. Thus, the connectivity condition (C) is not always satisfied.

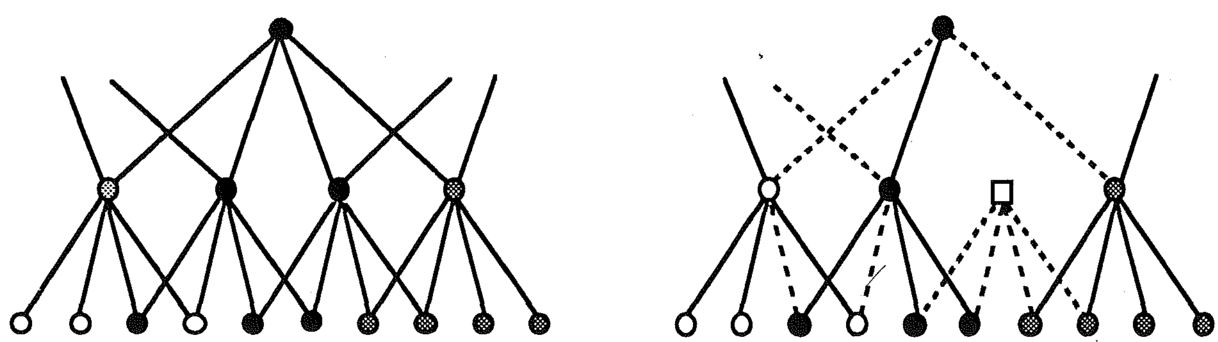

Fig. I.7.- Linked pyramid of a monodimensional signal

Besides the problem presented in Figure I.7, another main drawback of this method is that a stopping criterion has to be fixed before hand. If an external criterion is not imposed, such as the number of regions within the segmentation or the highest level in the pyramid at which a region can be found (region size), the algorithm may tend to build the pyramid up to the top (an example of this situation is given in [38]). Thus, a single region may be found as final segmentation.

The segmentation performed by utilising linked pyramids propagates information just vertically within the data structure (either upwards or downwards); that is, neither horizontal nor oblique links are built. In this way, the algorithm may overlook possible relationships between nodes at the same level that do not merge in a common father node (horizontal propagation), or between nodes at different levels in the decomposition (oblique propagation). This lack of information propagation translates into an oversegmentation of the image.

In the original work of linked pyramids [34], the homogeneity predicate relies on gray level homogeneity of the region, and the possibility of extending this criterion to more complicated predicates was just flashed out. Counting for this idea while trying to solve the above commented problems, some approaches have been described. In [39], a technique based on the linked pyramid data structure for segmenting textured images was proposed. In it, the homogeneity predicate uses more information than only gray levels. Another approach was developed in [40] which makes an extension for multidimensional data and allows propagation of the information all through the pyramid.

Although by means of these improvements very good segmentations are achieved, some problems still remain. That is, segmentations still present in some cases too many regions, and the way of controlling the amount of final regions is not straightforward. Furthermore, algorithms require more than a fair amount of time for performing the segmentation. This requirement is logical, since the algorithm has to go through the data structure several times before convergence and, afterwards, possible horizontal and oblique links have to be checked out.

I.5.- Structure and contributions

The structure of this work is as follows. After this introduction, Chapter II is devoted to the presentation of the main stochastic models used in image processing; namely, Gaussian Random Fields (GRFs), Markov Random Fields (MRFs) and Compound Random Fields (CRFs). GRFs are demonstrated to be a special case of MRFs. Furthermore, the equivalence between MRFs and Gibbs distributions is discussed. This equivalence allows algorithms to yield global maximum likelihood segmentations while handling only local computations. CRFs are defined as consisting of two different random fields, disposed in a hierarchical manner. This kind of image models is one of the mainstays of this work. The use of CRFs within the framework of image segmentation is described. Improvements achieved when using CRFs are highlighted. In addition, their main drawbacks are pointed out.

The three following chapters form the burden of this work. In them, the proposed segmentation technique is developed. In order to explained it, the following organisation is adopted. Chapter III describes the image model to be used. This image model utilises a second order, non-homogeneous Strauss process for characterising the boundary information and a set of white GRFs for the texture information. The reasons for choosing such a model are stated. Moreover, a monoresolution segmentation technique based on the above model and on a deterministic maximisation procedure is described. The performance of this technique is analysed as well as the influence of the selection of the model parameters on the final results. A set of parameters for the lower level model is selected. This selection is performed relying on the parameter robustness with respect to the final results. The dependence of the algorithm on the initial segmentation is also studied. This point is shown to be the most critical in the segmentation technique.

Chapter IV deals with the use of the previous segmentation technique in a multiresolution approach. A brief discussion on the advantages of multiple resolution decomposition in image analysis is carried out. The selection of the Gaussian and Laplacian pyramids as the multiple resolution decomposition to be used in this work is justified. A basic multiresolution segmentation scheme is presented. In this scheme, segmentation is performed at each decomposition level by means of the above monoresolution technique. The procedure starts segmenting the coarsest resolution level and its segmentation is used to guide the segmentation in the following finer level. This procedure is iterated down to the finest decomposition level. A first segmentation scheme is proposed in which the image model used at each level does not vary. Towards the goal of improving the segmentation performance, an unsupervised, adaptive scheme is presented. The model parameter adaptation is based on the information contained in the Laplacian pyramid. This second method is shown to yield good segmentation performance both in quality and computational load. However, the problem of correctly detecting interior regions is shown to be unsolved.

The problem of correctly segmenting interior regions is addressed in Chapter V. The problem is focussed from the viewpoint of discriminating between nearly flat, contrasted areas and rapid fluctuating, contrasted zones. Morphological tools are analysed under the scope of this problem, given that they are well known to efficiently perform such a task. The use of a morphological transform relying on the residue obtained with a morphological centre is proposed for extracting elements related to interior regions from the image. These elements are introduced in the multiresolution segmentation procedure as seeds for the obtaining of the interior regions. Several segmentation examples obtained by this final scheme are shown and discussed in detail.

In Chapter VI, the previous segmentation technique is used as basis for an image coding scheme. Block based and region based coding schemes are compared and the improvements achieved when coping with region based coding schemes are highlighted. A technique for coding the boundary information present in a segmented image is developed. A novel procedure for coding both shape and location of regions by means of chain code techniques is elaborated. Coding figures obtained by this scheme are shown.

Finally, a summary of conclusions is outlined in the last chapter. In addition, a set of possible future lines of research are suggested, regarding the segmentation technique, as well as the coding scheme.

Up to the best of our knowledge, the main contributions presented in this work can be summarised as follows:

- The use and analysis of an image model based on compound random fields for image segmentation. The lower level image model provides with a powerful tool for characterising the desired image boundary behaviour. Different boundary behaviours can be modelled by varying the lower level parameter values. The upper level level image model allows simple algorithms for estimating its parameters. This simplicity makes possible the updating of the model parameters with a feasible amount of computations.

- The proposition and performance study of a monoresolution segmentation technique relying on the previous image model and on a deterministic maximisation approach. This technique yields final segmentations always improving the quality of initial segmentations. In addition, it involves a small computational load.

- The presentation and analysis of a multiresolution segmentation technique based on the previous monoresolution method. The segmentation is performed by a top-down procedure. The use of such an approach reduces the likelihood of the algorithm to get trapped in local maxima. Coarsest level segmentations lead the algorithm to good quality segmentations at finest levels. The joint use of the information at different decomposition levels prevents the oversegmentation of textured areas. Furthermore, it has the additional advantage of reducing the total computational load with respect to the monoresolution approach.

- Improvement of the previous multiresolution technique by the adaptation of the image model to the data present in each level of the decomposition. The use of this adaptive procedure results in a better exploitation of the multiresolution analysis and yields high quality segmentations. Adaptation is carried out in a totally unsupervised way and it represents a negligible increase of the global computational load.

- The application of a morphological contrast extractor operator in a segmentation framework. This transform allows the detection of overlooked interior regions while discriminating between them and textured areas. Furthermore, it performs the extraction of connected image components related to interior regions. Such components are used as seeds in the previous multiresolution technique for correctly segmenting interior regions.

- The use of derivative chain codes and triple points information in a region based coding scheme. This scheme codes the region location information by introducing it in the chain codes describing the region shape information. This double use of chain codes reduces the amount of bits necessary to describe contour images. Huffman coding techniques are applied in order to further improve the coding performance.

I.6.- Summary

A definition of image segmentation has been given, making a stand that it needs to be vague, since the complexity of the segmentation task demands a such one. Therefore, and having in mind that for a given image several segmentations may accomplish this definition, a guide-line to recognise good segmentations is provided. Moreover, the desired behaviour of the ideal segmentation algorithm is described. In order to frame the problem, a set of applications has been listed. The different characteristics of these applications, as well as the intrinsic problems of segmentation, give an idea of its complexity. In spite of this complexity, a formal definition of segmentation is reported, which has been used throughout the sequel.

Once the problem has been stated, the rest of the chapter deals with a non exhaustive overview of the classical worklines in the segmentation framework. Firstly, edge-oriented techniques are summarised. It has been emphasised that complex algorithms must be applied in order to obtain closed boundaries (which is a main requirement in the segmentation definition). By using thresholding techniques, it is shown that good results can be achieved only in the case of very simple images. This drawback is owing to the fact that thresholding algorithms make use of only gray level criteria, while withdrawing more complicate homogeneity notions as well as the connectivity aspect.

Mixing homogeneity and connectivity concepts, region growing techniques lead to better results. However, in order to ensure this quality, the use of model-based criteria for the homogeneity predicate is mandatory. Furthermore, possible mergings must be sorted for turning the segmentation independent of its initial conditions. These two constraints make these algorithms very time demanding and, therefore, not very popular. Moreover, region growing is a bottom-up technique which leads to oversegmented results in case of textures or complicated images. Due to these drawbacks, it is necessary to seek new solutions.

Towards this goal, "split and merge" techniques approach the problem from a mixture of top-down and bottom-up viewpoints. These methods can still deal with model-based predicates, while using a multiresolution analysis of the images. Nevertheless, this analysis is not completely exploited and the method does not rely solely on it. Therefore, new problems arise: segmentations present "block" effects and more than a data structure is necessary.

In order to solve these problems, and taking the uncertainty concept as core idea, linking pyramid segmentation has been introduced. The advantages of multiresolution decomposition are fully exploited and regions satisfying any possible shape are allowed. However, the control of the characteristics of the final segmentation (connectivity of regions and their number, mainly) is not easily performed. In addition to that, methods relying on the linking pyramid data structure require a fair amount of computations, which makes them very slow.

In conclusion, although some steps towards the desired segmentation algorithm have been taken and some of its necessary characteristics seem clear (e.g.: the use of elaborate models and of multiresolution schemes), there are still plenty of tasks to perform and more than a handful of points to clarify. In the sequel, some new approaches to face the segmentation problem, which solve some of the above problems, will be presented.