Hierarchical Region Based Stereo Matching

Laurent Cohen   Laurent Vinet   Peter T. Sander
André Gagalowicz
INRIA
Domaine de Voluceau-Rocquencourt, B.P.105
78153 Le Chesnay Cedex, France
tel: 39 63 53 66

March 28, 1995

Proceedings of Computer Vision and Pattern Recognition
San Diego,June 1989

Abstract

Stereo matching is the process of determining correspondences between entities in related images. Often, this is treated as two quite independent subprocesses: segmentation, followed by matching. In this paper, we treat these processes as naturally related, in that partial matching results are fed back to the segmentation and both proceed simultaneously in a cooperative fashion. We consider regions as the primitives to be matched, since we feel that many of the shortcomings inherent in approaches based on points or lines can be overcome by taking more developed entities. Our implementation is based upon maintaining a hierarchy of segmented regions in each image, corresponding to analysis at differing scales. The selection of a particular segmentation in each image at a scale appropriate to each region is validated with reference to the optimal matching region in the other image. We present examples of our methods applied to a synthetic image (incorporating colour), and to natural office scenes.

1 Introduction

Vision enables a system to interact richly with its environment. A fundamental task solved with facility by biological systems is the visual discrimination of objects and their situation (shape/location) in space, and one of the strategies evolved for the job is stereo vision. Similarly, stereo is one of the methods of choice for equipping autonomous robots with visual perception. In this paper we present a novel approach to the combined problem of image segmentation and object distance computation based on interaction between a segmentation component and a stereo component. We believe the combination to be better than the parts taken individually.

Two distinct subproblems are involved in the determination of the distance of an object by means of stereo: the matching problem and the depth computation proper. Given a stereo pair of images, the matching problem consists of associating entities from the left image with entities in the right image.

Stereo matching with points and lines as the entities has become a well developed industry. In this paper, we investigate region based matching as we feel that many of the shortcomings inherent in other approaches can be overcome by taking more developed entities. To cite but two examples: mismatches over pairs of line elements are to be expected frequently due to the lack of features available for distinguishing between segments; and occlusion effects are relatively more severe when applied to points or segments than to regions. We consider only the matching problem—we are not concerned with the depth computation here, but see [12].

The quality of discreteness of regions is determined by the segmentation of the image. Since no satisfactory method of deriving 'the' segmentation yet exists, we do not commit ourselves rigidly to any particular segmentation prior to the matching process. That is, segmentation and matching are not independent sequential processes, but rather, partial matching results are fed back to the segmentation and both proceed simultaneously in a cooperative fashion. We make minimal use of special-purpose *a priori* information about, for example, image formation and object formation, but make good use of available information by considering the segmentation of the pair of images together. While we effect stereo matching over homogeneous regions, we incorporate information about discontinuities by integrating edge detector results into our region segmentation algorithm. The problem of obtaining a semantically valid segmentation (one whose regions correspond to perceptually meaningful entities) by simple homogeneity measures over groups of pixels remains open.

The basic idea developed in this paper is that, since objects in the world being imaged give rise to events in both stereo images (modulo occlusions and border effects), segmentation in each image should be carried out in conjunction with segmentation in the other, thus, hopefully, producing a more reliable segmentation in both. Of course, a 'vicious circle' can arise in that cooperative segmentation presupposes matching, and matching is dependent on a prior segmentation. We propose breaking the circle by iteratively using partial segmentation results to suggest tentative matches, which then feed back into the segmentation procedure, and so on.

The paper is organized as follows. The following section briefly presents some related work in image segmentation and stereo matching, and §2 gives an overview of our methods. Section 3 is the segmentation component. Section 4 is the region-based matching algorithm, which explains the feedback of matching to segmentation, and presents our results.

1.1 Related work

There exists a natural complementarity between edge based methods and region based methods for image segmentation. Region based methods seek homogeneity among pixels according to certain criteria (generally based on grey level statistics). Pixels which satisfy given criteria are grouped together into regions on the assumption that intra-object grey levels are approximately constant. A popular region segmentation method is the quadtree based split-and-merge algorithm [14] and its variants (see [18] for an early survey). The resultant square blocks of pixels are generally merged with adjacent blocks on the basis of homogeneity criteria to produce the final segmentation into irregularly shaped regions.

Image segmentation by region growing using multiple predicates has been proposed by [11], although for a single level only.

Not much has been published on the integration of edge detection and region growing techniques, although see [6, 8].

In [16], we can find some mathematical basis for the use of region growing techniques using homogeneity predicates.

Stereo matching has been done on the basis of the raw image grey levels by correlation techniques [9], and by matching entities or features extracted from the images separately (we refer the reader to the surveys [2, 10]). The most commonly used features are points representing estimated edge elements. Line elements may also be used [1]; the only references we are aware of for the use of regions as features are [4, 5].

In most approaches, with the exception of 'stereo snakes' [15], feature extraction proceeds independently of and preceeds the stereo matching. In contrast, our method depends in an essential fashion on the interaction between segmentations in both the stereo images and matching between them.

2 Overview

Here, we present a brief overview of our system; following sections fill in the details.

As mentioned above, our aim is for segmentation to proceed cooperatively with stereo matching. Segmentation by merging and splitting regions in the left image should depend on what matches have been found with the right image, and vice-versa. Some of the computation can be done independently, however, prior to any matching. In particular, if a number of (candidate) segmentations of the images are computed for a range of parameters and organized in a tree structure, then merging/splitting regions just amounts to moving up/down in the tree. Thus, the complete procedure consists of two steps:

1. computing, independently for each image, fine to coarse hierarchical candidate segmentations;
2. determining a final segmentation, choosing for each pixel the most appropriate region level among the candidates, cooperatively with region based stereo matching between the images.

2.1 Segmentation

Let us recall what we mean by a segmentation $S = \{R_1, R_2, \ldots\}$ of a set E defined by a predicate P (as in [14]):

1. S is a partition of E,
2. $P(R_i)$ is true for all i,
3. if $i \neq j$ then $P(R_i \cup R_j)$ is false.

A hierarchical segmentation is a sequence $S_0, S_1, \ldots, S_n$, where each level S_i is a segmentation defined by predicate P^i and which contains the previous S_{i-1}, i.e., $\forall R \in S_{i-1}$, $\exists \bar{R} \in S_i$ such that $R \subset \bar{R}$. Note that each segmentation level may result from the successive application of several predicates, P^i_j, $j = 1, 2, \ldots, n_i$, say.

The pseudo code in Fig. 1 gives the organization of the segmentation step. The outer `while` loop computes (potential, or candidate) segmentations for an entire range of parameter values in both images, arranged in the form of two trees (hierarchical graph structures). Level 0, at the bottom of the hierarchy, consists of fine segmentations, i.e., small regions, with increasing levels producing progressively larger regions.

```
set lowest level segmentation parameters;
while (segmentations halt criterion not satisfied)
      set initial predicate P;
      while (not all predicates already applied)
            compute MC, list of pairs of adjacent regions
              ordered by cost of merging according to P
            while (MC not empty)
                  if (P(regions of head of MC) is true)
                     merge regions;
                  MC ← tail of MC;
            set next predicate P;
      set next segmentation level parameters;
```

Figure 1: The organization of the segmentation process.

The middle `while` loop indicates that various predicates determine the merge criteria at each level, and the predicates are applied to produce merges pairs of adjacent regions in the inner loop. An example of such a predicate might be based on the mean grey-level intensities of regions

$$P_{\text{mean}}(R_i, R_j) \equiv (|\text{mean}(R_i) - \text{mean}(R_j)| < t_{\text{mean}})\,.$$

A segmentation depends on the order of the merges. To avoid having the order depend on the image traversal strategy, obviously unsatisfactory, we carry out the merges in order of increasing cost, according to the appropriate predicate.

2.2 Cooperative matching

We stress the distinction between computing the graphs representing multiple levels of segmentation of the images (step 1 above), which is done independently in each image, and commitment to a particular set of regions as the resulting segmentation (step 2), which is done *while* stereo matching. Step 2 of the process then amounts to computing node cut sets [18], see Fig. 2, through the graphs by mapping pixels to nodes, and this is done cooperatively with the partial results of stereo matching.

The region based stereo matching associates regions in the left graph with regions in the right which are likely to be images of the same physical object. Since image formation parameters can differ, the same segmentation parameter is not guaranteed to give similar results in both images. Thus, matching may occur across levels of segmentation. The algorithm appears in Fig. 3.

Beginning at the the top (coarsest) level of segmentation, region L of the left image matches region R of the right whenever

$$\max_{\hat{R}\in\Lambda} s(L,\hat{R}) = R,$$

where Λ is the set of regions of the right image eligible to match L, and $s(L,R)$ a measure of similarity between regions (the precise formulation is in §4). If L at some level fails to find its match in the other image, then its descendents in the segmentation hierarchy are added to the list of regions to be matched. Whenever a match is found, then both matched regions and all their descendents are no longer considered open for matching.

3 Segmentation

Edge detection and region growing are two intimately related aspects of image segmentation, yet are rarely used together. In this paper, we exploit their natural complementarity to enhance the segmentation.

A fundamental tenet of this paper is that segmentation in the pair of stereo images should be done cooperatively, this is, segmentation in the left should take into account segmentation in the right, and vice-versa. In the interests of algorithm efficiency, we pre-compute segmentations at various granularities and store them in a hierarchical region adjacency graph structure [13, 11]. Thus, for a region at a given level, splitting it into subregions or merging it with other regions just involves changing levels in the graph structure.

We present here the creation of the segmentation hierarchies, which can be carried out independently in each image. All segmentation levels are considered equally valid in that we make no decision here as to which level segmentation a pixel belongs. As described in §4, it is the interaction between images which decides the ultimate segmentation.

[]done on Mac

Figure 2: Schematic diagram of the hierarchical region graph structures. The sequence of segmentations for the left image shows the parent-child relationship between regions at different levels. The arrows between regions of the left and right images shows that matching may take place between different levels.

```
set 𝓛 = {regions of coarsest segmentation of left image};
set 𝓡 = {regions of coarsest segmentation of right image};
while (𝓛 or 𝓡 has changed)
      for (every region in 𝓛)
          determine eligible regions in 𝓡;
          for (every eligible region R)
              compute similarity s(L,R);
      while (max_{L∈𝓛, R eligible to match L} s(L,R)) sufficient
          match L,R;
          remove L and all relatives from 𝓛;
          remove R and all relatives from 𝓡;
      for (every L ∈ 𝓛)
          add descendents to 𝓛;
      for (every R ∈ 𝓡)
          add descendents to 𝓡;
```

Figure 3: The organization of the process to find a match in the right image for regions of the left image. Everything is similar for regions of the right image.

3.1 Edge detection

Edge detection estimates how strongly image pixels correspond to regions of intensity change, based on the assumption that such changes in the image correspond to relevent physical events. The Canny edge detector [3], approximated as the first derivative of a Gaussian, satisfies the three desirable criteria of low probability of missing an edge, good localization, and low probability of multiple responses per edge. We use an improved implementation of this edge operator [7] from which to estimate image contours. The contours act as barriers in the merging process: neighbouring pixels are prevented from merging across contours, see §3.2.3. Figure 4 shows a simple example of how edge information can improve the segmentation.

3.2 Region growing

Both images are segmented independently into a hierarchy of candidate, or potential, segments, but with no commitment to any particular one. Thus, we

without contours: i004g_s2
with contours: i002g_s2
contours: i002g_c
or the synthetic glass

Figure 4: Comparison of the use of edges in the segmentation algorithm. top: (left) original; (right) without using edges. bottom: (left) with exclusion of edge pixels, note the green region at the bottom which contains long edges in its interior; (right) avoids passing through edges for merging two regions.

produce fine to coarse segmentations $S_0, S_1, \ldots, S_n$. Segmentation proceeds 'upwards' (fine to coarse) from an initial level by merging neighbouring regions satisfying homogeneity conditions (§3.2.3).

3.2.1 Initialization by quadtree operations

The region merging algorithm, described in the next section, may begin with pixel-sized regions. For reasons of efficiency, however, we begin with initial regions created by standard quadtree operations [17]. This simple pre-processing allows a substantial reduction in the number of initial regions.

Essentially, we utilize the first part of the merge algorithm in [14]. Let the image pixels be (ij) and define square blocks of four 'pixels' at various levels by

$$\begin{aligned} B^0_{i,j} &= (ij) \\ B^l_{i,j} &= B^{l-1}_{i_1,j_1} \cup B^{l-1}_{i_1+1,j_1} \cup B^{l-1}_{i_1,j_1+1} \cup B^{l-1}_{i_1+1,j_1+1}, \end{aligned}$$

where $i_1 = 2i, j_1 = 2j$.

Then the four blocks $B^{l-1}_{i_1,j_1}$, $B^{l-1}_{i_1+1,j_1}$, $B^{l-1}_{i_1,j_1+1}$, $B^{l-1}_{i_1+1,j_1+1}$ are considered merged into $B^l_{i,j}$ whenever

$$\max_{B^l_{i,j}} I(i,j) - \min_{B^l_{i,j}} I(i,j) < t^0_{\mathbf{minmax}},$$

for some appropriate threshold of the image intensities $I(i,j)$. Note that contour information (see §3.2.3) is already incorporated at this level, so that blocks are not merged when they are separated by sufficiently strong edge elements. Figure 5 shows a typical region initialization produced by the quadtree merges.

i002g_q

Figure 5: top: original stereo pair;bottom: quadtree using contours.

3.2.2 Levels of segmentation

Since we haven't found a principled way to set parmeters to produce 'the' segmentation, we defer committing ourselves to a particular segmentation at this stage, and generate instead a hierarchy of segmentations at various thresholds. These are computed independently in each image, and are to be taken as a pre-processing step. The determination of the actual segmentation occurs in conjunction with stereo matching, and merging/splitting regions is then equivalent to moving up/down in the region hierarchy.

We begin with the square quadtree regions output from the initialization, and choose fairly selective parameters t_k^0 (i.e., permitting only the most obvious region merges). The initialization is not a segmentation in the sense of §2.1 since it is not maximal. That is why the first level considered is the segmentation completed from the initialization using some predicates P_k^0.

The parameters are then progressively relaxed, as in the outer `while` loop of Fig. 1, permitting more permissive merges, and the resolution of the segmentations of the hierarchy moves from fine to coarse. For each characteristic k, the progression of thresholds $t_k^0 < t_k^1 < \cdots < t_k^n$ controls the shape of the segmentation graph, and is such that the levels become finer towards the top. Generally, the ultimate t_k^n are taken to be very large to permit all possible region merges.

Table 1 shows the organization of the parameters of the various segmentation levels. Note that each level is created by the application of multiple merge predicates to pairs of adjacent regions, as in the middle loop of Fig. 1.

predicate	threshold	segm level	
P_0^n	t_0^n		
$\vdots$	$\vdots$	n	coarse
P_m^n	t_m^n		
$\vdots$	$\vdots$	$\vdots$	
P_0^1	t_0^1		
$\vdots$	$\vdots$	1	$\vdots$
P_m^1	t_m^1		
P_0^0	t_0^0		
$\vdots$	$\vdots$	0	fine
P_m^0	t_m^0		
P^q	t^0	quadtree	initialization

Table 1: Successive predicates create various segmentation levels. The quadtree level is the initialization, zero is the finest level, n the coarsest.

3.2.3 Merge conditions

The result of the initialization by quadtree merging is to segment the image into square regions satisfying intensity homogeneity conditions. Grouping next considers adjacent regions (rather than regions with a common quadtree parent as in the initialization). Adjacent regions R_i^l, R_j^l are merged into one region whenever the predicate P_k^l is true, where, given a cost-of-merging function C_k^l,

$$P_k^l(R_i^l, R_j^l) \equiv \left(C_k^l(R_i^l, R_j^l) < t_k^l\right).$$

Multiple merging predicates may be successively applied at each level, as in Table 1. Our criteria are derived from simple image statistics, e.g., P_{minmax}^l and P_{mean}^l are based on the cost functions

$$\begin{aligned} C_{\text{minmax}}^l(R_i^l, R_j^l) &= \max(R_i^l \cup R_j^l) - \min(R_i^l \cup R_j^l) \\ C_{\text{mean}}^l(R_i^l, R_j^l) &= |\text{mean}(R_i^l) - \text{mean}(R_j^l)|. \end{aligned}$$

We consider edge elements as 'special regions' with the following properties

1. edge regions merge to other edge regions (to create linked edges), but cannot merge to 'normal' regions;

2. an edge region separating two regions can prevent the merging of those regions.

Thus, a merge between adjacent R_i^l, R_j^l is considered only if P_{edge}^l is true,

$$P_{\text{edge}}^l(R_i^l, R_j^l) \equiv \left(\frac{\text{edge\_length}(R_i^l, R_j^l)}{\text{frontier\_length}(R_i^l, R_j^l)} < t_{\text{edge}}^l\right),$$

where frontier_length is the boundary length between the two regions (the number of pixels where the regions are adjacent), and edge_length is the number of actual edge pixels that separate the two regions (pixels of edge regions that are adjacent to both regions).

If we did not use this criteria, two similar regions (in the sense of predicates) that are separated by an edge region and that have a small frontier_length would be merged, leaving the edge region isolated inside the new region.

The grain of edge elements should be appropriate to the grain of the regions. As the segmentation into regions becomes coarser, weak edges are converted into normal regions and disappear by becoming merged into larger regions. Thus, at the coarsest segmentations, only strong edges remain to constrain the merges.

3.2.4 Merge ordering

The order in which pairs of regions are merged has been shown to influence the results of merging algorithms [18]. Thus, the inevitable order dependence must be motivated by something more rigourous than just the image traversal strategy. We first associate a cost to merging each mergeable pair of adjacent regions, and then merge regions in order of increasing cost. Once the list of merge costs has been exhausted (perhaps up to some threshold), there is no more to be done with the current predicate. We then consider the next predicate, or, if all predicates have been applied to this level, relax the segmentation parameters to permit more permissive merges, and carry out the above process at the new segmentation level. This is repeated until the cost of region merges becomes prohibitive.

The resulting segmentations at various granularities are shown in Fig. 6.

[]i002g_s[0...3]

Figure 6: Levels of segmentation of the left image (fine to coarse from upper left to lower right).

4 Region based stereo matching

The region based matching procedure exploits the hierarchical region graph described in the previous section. It is during this matching process that we make a committment to a particular segmentation level for each region. Recall that the creation of the segmentations is effectively just a pre-processing step and doesn't change the fundamentals of the algorithm. Contrary to the segmentation, which proceeds bottom-up (fine regions to coarse), matching begins at the top of the segmentation tree and works downwards. This makes better use of larger regions where the matches are expected to be more reliable.

Let $\mathcal{L}, \mathcal{R}$ be the sets of all the regions at the top (level n) of the segmentation structures of the left and right images respetively. Given region L_i^n in the left image, we consider a set of regions $\Lambda_i^n \subset \mathcal{R}$ of the top level of the right image which are *admissible* matches to L_i^n. The set Λ_i^n could, in principle, be the entire $\mathcal{R}$, but when we are given the geometry of the cameras, we can restrict Λ_i^n to regions whose centre of gravity is 'close' to the epipolar of the centre of gravity of L_i^n. In addition, we further restrict the regions of Λ_i^n by imposing rough size-similarity (based on the number of pixels) and circularity (based on the first moments) constraints relative to L_i^n.

For each $R_j^n \in \Lambda_i^n$, we then compute a measure of overall similarity

$$s(L_i^n, R_j^n) = \sum_{p=1}^{q} w_p s_p(L_i^n, R_j^n),$$

for weight w_p and various resemblance functions between regions

$$s_p(L, R) = 1 - \frac{\min(A_p(L), A_p(R))}{\max(A_p(L), A_p(R))}.$$

A_p is some attribute of a region, for example, intensity mean, intensity variance, spatial moment, etc. All pairs of matchable regions are stored in list form by order of decreasing similarity. Note that the left region L_i contributes a pair to the list for each element of Λ_i, and that these pairs are not necessarily contiguous on the list since they are ordered by similarity. Matching then proceeds simply down the ordered list of similar pairs. Once a region finds a match, any other pairs of which it is a member are henceforth ignored, since their constituents are, by construction, less similar. Pairs are considered in order and removed from the list until the measure of similarity between the next pair falls below a given threshold.

It is at the moment of matching that we finally make a definitive committment to a particular segmentation. Only when a region is finally matched, do we consider that its pixels constitute a region in the sense of the final segmentation. If it happened that all regions were matched at the coarsest level, that is, all the measures of similarity were sufficient, there would be no reason to go further and we would consider it the segmentation. This is (unfortunately) unlikely to occur, hence we proceed iteratively, downward in the tree.

All regions which remain unmatched are split, that is, their children (previously computed) are all added to the region lists $\mathcal{L}, \mathcal{R}$ and participate in the further matching. These regions now undergo exactly the matching process described above. With the inclusion of a level of children, inter-level matching becomes possible: $s(L_i^n, R)$ may indicate more similarity when R belongs to some level other that n. Each iteration descends one level in the segmentation graph and adds children regions from the new region to the sets of matchable regions. When the iteration is carried to the limit, it leads to testing the matchability of each unmatched region in the left image to each unmatched region at any level in the right. Note that this is not the same as testing all regions against all other regions, with its potential for combinatorial explosion, since regions are eliminated from consideration once they become matched (along with their parents and children).

Matching stops when there is nothing left to do, when no remaining pair of admissible matches is sufficiently similar (and this is guaranteed to take place, since there are finitely many regions and some are eliminated from consideration at each iteration). It is also only now that we consider a final segmentation to have taken place through the interaction due to the matching component between the left and right potential segmentations. It may well be that the resultant segmentations are incomplete in that not every image pixel is assigned to a particular region, since not every region can necessarily be expected to find a match. However, we have found that leaving some regions unmatched does not detract from the overall quality of the results. It seems, in fact, preferable to accept only reliable matches than to force the maximum number of matches and accept matches of dubious quality.

5 Conclusions

Our approach to stereo image analysis, presented in this paper, is based on three tenets, which address the basic problem of how to make use of as much image information as possible. First, image segmentation and matching should not be independent successive processes. There is information in each image relevant to the analysis of the other, and this should be incorporated into the segmentation as well as the matching step. Second, regions possess more structural information which is stable to small changes of viewpoint than do edges or points. Hence, we expect to make more stable matches by taking regions as the primitive elements. Third, and related to the previous point, edge- and region-based methods are naturally complementary, and should be used together for segmentation; neither should be considered as an end in itself. We have developed programs to test these assumptions, and we feel that the results are indeed promising.

au choix: i00[1-9][gd]_m*

Figure 7: Top: Matching regions. Bottom: resultant segmentation.

Acknowledgements

Rachid Deriche provided an implementation of the Canny edge operator. Francis Ledru created the synthetic office scene, and Olivier Faugeras supplied the stereo pair of the office scene.

References

[1] Nicholas Ayache and Francis Lustman. Fast and reliable passive stereovision using three cameras. In *International Workshop on Industrial Applications of Machine Vision and Machine Intelligence*, Tokyo, February 1987.

[2] Stephen T. Barnard and Martin A. Fischler. Computational stereo. *Computing Surveys*, 14(4):553–572, December 1982.

[3] John Canny. A computational approach to edge detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI-8(6):679–698, November 1986.

[4] Jean-Pierre Cocquerez and André Gagalowicz. Mise en correspondence de régions dans une paire d'images stéréo. In *Machines et Réseaux Intelligents*, Paris, May 1987.

[5] Jean-Pierre Cocquerez and Olivier Monga. Matching regions in stereovision. In *Proceedings of the Fourth Scandinavian Conference on Image Analysis*, Stockholm, 1987.

[6] Edgar A. Cohen, Jr. Generalized sloped facet models useful in multispectral image analysis. *Computer Vision, Graphics, and Image Processing*, 32:171–190, 1985.

[7] Rachid Deriche. Using Canny's criteria to derive a recursively implemented optimal edge detector. *International Journal of Computer Vision*, pages 167–187, 1987.

[8] Rachid Deriche and Jean-Pierre Cocquerez. Extraction de composantes connexes basee sur une detection optimale des contours. In *Actes, Machines et Reseaux Intelligents, Cognitiva 87, Image Electronique*, volume Tome 2, Paris, Mai 1987.

[9] Richard O. Duda and Peter E. Hart. *Pattern Classification and Scene Analysis*. Wiley, New York, 1972.

[10] Olivier D. Faugeras. A few steps toward artificial 3 d vision. Technical Report 790, INRIA, February 1988.

[11] André Gagalowicz and Olivier Monga. A new approach to image segmentation. In *Proceedings of the Eighth International Conference on Pattern Recognition*, Paris, October 1986.

[12] André Gagalowicz and Michel Peyret. Reconstruction 3d basée sur une analyse en régions d'un couple d'images stéréo. In *Proceedings of Pixim*, Paris, October 1989. In preparation.

[13] J.J. Gerbrands and E. Backer. Split-and-merge segmentation of SLAR-imagery: Segmentation consistency. In *Proceedings of the Seventh International Conference on Pattern Recognition*, pages 284–286, Montréal, July 1984.

[14] S.L. Horowitz and T. Pavlidis. Picture segmentation by a directed split-and-merge procedure. In *Proceedings of the Second International Joint Conference on Pattern Recognition*, pages 424–433, 1974.

[15] Michael Kass, Andrew Witkin, and Demetri Terzopoulos. Snakes: Active contour models. In *Proceedings of the First International Conference on Computer Vision*, pages 259–268, London, June 1987.

[16] Jean-Michel Morel and Sergio Solimini. Segmentation d'images par méthode variationnelle: une preuve constructive d'existence. *Comptes Rendus de l'Académie des Sciences*, 1988.

[17] Hanan Samet. The quadtree and related hierarchical data structures. *Computing Surveys*, 16(2):187–260, June 1984.

[18] Steven W. Zucker. Region growing: Childhood and adolescence. *Computer Graphics and Image Processing*, 5:382–399, 1976.