Hierarchical Region Based Stereo Matching

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

Abstract

Often, stereo matching is treated as two quite independent subprocesses: segmentation, followed by matching. In this paper, we treat these processes as naturally related, with partial matching results feeding back into the segmentation and both proceeding simultaneously in a cooperative fashion. We consider regions as the primitives to be matched, and 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 for a region at an appropriate scale in one image is validated with reference to the optimal matching region in the other image. We present examples of our methods applied to image of real 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.

Stereo matching with points and lines as the entities has become a well developed industry. 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.

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. Some of the computation can be done independently, however, prior to any matching. 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 fine to coarse hierarchical candidate segmentations, independently for each image;

2. determining a final segmentation by choosing for each pixel the most appropriate region level among the candidates, cooperatively with region based stereo matching between the images.

1.1 Related work

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 homogeneous. A popular region segmentation method is the quadtree based split-and-merge algorithm [1] and its variants (see [2] 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.

In [3], we can find some mathematical basis for region growing techniques using homogeneity predicates. Image segmentation by region growing using multiple predicates has been proposed by [4], although for a single level only.

Stereo matching has been done on the basis of the raw image grey levels by correlation techniques, and by matching entities or features extracted from the images separately (we refer the reader to the surveys [5, 6]). The most commonly used features are points representing estimated edge elements. Line elements may also be used [7]; the only references we are aware of for the use of regions as features are [8, 9].

In most approaches, 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 Segmentation

We present here the creation of the segmentation hierarchies, step 1 above, 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 §3, it is the interaction between images which decides the ultimate segmentation.

A predicate P defines a segmentation $S = \{R_1, R_2, \ldots\}$ of a set E when [1]

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 below 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. 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.

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 associated predicate.

```
initialize regions;
initialize (level $l = 0$) segmentation parameters $t_1^l, \dots, t_m^l$;
while (segmentation halt criterion not satisfied)
  initialize ($k = 0$) cost function $C_k^l$ and predicate $P_k^l$;
  while (not all predicates already applied)
    compute list of costs of merging adjacent regions
      MC $= \{C_k^l(R_{i_0}, R_{j_0}) \leq C_k^l(R_{i_1}, R_{j_1}) \leq \cdots\}$;
    while (MC not empty)
      if ($P_k^l$(regions of head of MC) is true) merge regions;
      MC $\leftarrow$ tail of MC;
    next predicate $P_{k+1}^l$;
  next segmentation (level $l + 1$) parameters;
```

2.1 Region growing

We produce segmentations $S_0, S_1, \dots, S_n$ proceeding 'upwards' (fine to coarse) from an initial level by merging neighbouring regions satisfying homogeneity conditions (§2.1.2).

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 [10]. This simple pre-processing allows a substantial reduction in the number of initial regions.

i002g_o

Figure 1: original stereo pair.

2.1.1 Levels of segmentation

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 parameters are then progressively relaxed, 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.

2.1.2 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

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

Table 1: Successive predicates create various segmentation levels.

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

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

respectively.

We consider edge elements as ‘special regions’ with the following properties: edge regions merge to other edge regions (to create linked edges), but cannot merge to ‘normal’ regions; 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.

2.1.3 Merge ordering

The order in which pairs of regions are merged has been shown to influence the results of merging algorithms [2]. 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. 2.

[]i002g_s[0...3]

Figure 2: Levels of segmentation of the left image (fine to coarse from upper left to lower right).

3 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, step 2 in 1, 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. 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.

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, see the following pseudo-code. 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.

```
$\mathcal{L}$ initialized to $\{L_i^n\}$;
$\mathcal{R}$ initialized to $\{R_j^n\}$;
do
  for (all $L_i \in \mathcal{L}$)
    determine eligible regions $\Lambda_i \subset \mathcal{R}$;
    for (all $R_j \in \Lambda_i$)
      compute similarity $s(L_i, R_j)$;
  while ($\max_{L_i \in \mathcal{L}, R_j \in \Lambda_i} s(L_i, R_j)$) sufficient
    match $L_i, R_j$;
    remove $L_i$ and all relatives from $\mathcal{L}$;
    remove $R_j$ and all relatives from $\mathcal{R}$;
  for (all $R \in \mathcal{L} \cup \mathcal{R}$)
    add descendents to $\mathcal{L}$ or $\mathcal{R}$;
while $\mathcal{L}$ or $\mathcal{R}$ has changed;
```

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.

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

Figure 3: Top: Matching regions. Bottom: resultant segmentation.

4 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.

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] 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.

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

[3] 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.

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

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

[6] Olivier D. Faugeras. *A Few Steps toward Artificial 3 D Vision.* Technical Report 790, INRIA, February 1988.

[7] 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.

[8] 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.

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

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