SEGMENTATION ALGORITHM BY MULTICRITERIA REGION MERGING

B. MARCOTEGUI
Centre de Morphologie Mathématique, Ecole des Mines de Paris
35, rue Saint-Honoré, 77305 Fontainebleau Cedex, France
tel : 64 69 47 06 Email : marcotegui@cmm.ensmp.fr

Abstract. This paper presents a segmentation algorithm for image sequences, by multicriteria region merging. The output of a connected filter is simplified by iteratively merging the two most similar adjacent regions, while a given representation quality is preserved. We have defined several region-similitude criteria: grey-level, texture and motion resemblance. Texture and motion criteria introduce a feedback "segmentation-coding step" that improves coding efficiency.

Key words: segmentation, region merging, image sequences, image coding

1. Introduction

An object-oriented image compression system splits up the image to be coded into regions that correspond as much as possible to the objects present in the scene. Then, the contours of the partition and the content of each region are separately coded.

Usually segmentation and coding steps are two independent stages, which can lead to two undesirable situations:

- the partition contains several regions that would have been correctly coded as one region with a lower coding cost;
- the partition is too poor and the texture coding does not represent correctly the content of the regions.

The algorithm presented in this paper introduces a feedback between the segmentation and the coding steps, improving the codec efficiency. The idea is to merge the regions that are satisfactorily represented together. The gain in coding cost comes from the fact that:

- only one set of texture or motion parameters is used to describe the resulting region;
- contour pixels (which must be coded) "disappear" in the merging process.

Thus, the coding cost is reduced while the visual quality is not significantly affected.

We have developed a region merging algorithm, able to use various types of criteria. Several merging criteria, taking into account coding information, have been integrated.

The algorithm is applied to image sequences and we can distinguish two different working modes:

- INTRA mode, or initializing mode. The whole scene is unknown and must be entirely coded.

- INTER mode, only changes with respect to the previous time are transmitted. The evolution of the objects of the scene are coded by means of a motion vector, next the prediction error is corrected.

We will refer to both modes in the following.

Section 2 describes the implemented region merging algorithm. Section 3 describes the conditions in which the algorithm is applied. Section 4 presents the different merging criteria we have developed. The results are illustrated in section **??**.

2. Region Merging Algorithm

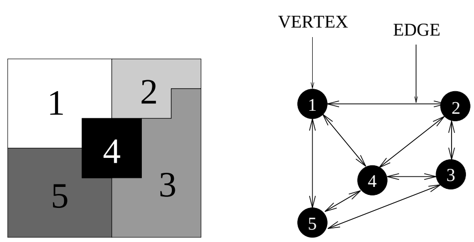
The flat-zones (regions) of an image are the largest connected components of pixels possessing the same function value. An original image contains a large amount of small flat-zones and segmentation consists in producing larger zones in smaller number.

The main characteristic of connected operators [**?**] is that they process flat-zones instead of isolated pixels. This has two major advantages:

- The merging of flat zones removes existing contours but never generates new ones. This fact allows the preservation of the object shapes.
- After a certain number of fusions, based on a simple criterion such as contrast, the resulting regions may accept more complex attributes than pixels, for example a texture or a motion representation.

The region merging algorithm consists in iteratively merging the two most similar adjacent flat-zones until a stop criterion is reached (similitude measure and stop criterion will be defined in the following). These successive fusions define a series of partitions that describe the image with a decreasing precision degree: it is a bottom-up segmentation algorithm.

The algorithm needs to handle easily all the possible fusions of an image. It can be implemented in an efficient way with a graph structure whose vertices and edges are valuated. When associating a graph to a segmented image, vertices represent the segmented regions and edges the neighborhood relations among them. Figure. 1 shows the relationship between images and graphs.



(a) Input Image (b) Associated graph

Fig. 1. Images and Graphs

Edges correspond to the pairs of adjacent regions. Their valuations, which correspond to a measure of similarity between two adjacent regions, provide the merging priority: the edge of lowest valuation will be the first edge to be removed because it separates the two most similar regions.

The algorithm can be split up into the following steps:

- Compute the graph associated with the input image
- Valuate the edges of the graph according to a merging priority
- Find the lowest valuation edge and:
  = merge the regions corresponding to the edge, that is, remove the edge from the graph
  = update the similarity measure of edges around the resulting new region
  Iterate this point until reaching the stop criterion
- Generate the image associated with the new graph

Hierarchical queues [**?**] are used to store the ranking order of edges so that they can be efficiently accessed in increasing order of their valuation.

The region merging algorithm needs a stop criterion in order not to reduce the image to one region. This criterion may be one of the following: a number of output regions, a number of output contour points, a compression rate (a combination of the number of regions and of the number of contour points), another criterion (depending on the merging criterion).

3. Application of the algorithm

3.1. Pre-processing

The algorithm applied directly to an original image has two major drawbacks:

- it leads to an expensive implementation in terms of computation time.
- it preserves a lot of regions of only one pixel that do not correspond to important details of the image but to noise or to strong transitions.

Both drawbacks are avoided by means of a pre-processing step. We use connected filters of Mathematical Morphology (i.e. filters by reconstruction, area filters [**?**]...) that remove small details (essentially noise) producing flat-zones. Since these filters only act on the extrema of the image, very small flat-zones remain in transition areas; they are then assigned to larger adjacent flat-zones, on the basis of a contrast criterion. The result of this procedure is a first partition with a large number of regions.

3.2. Processing in "intra"-mode

The region merging algorithm produces an intra-segmentation avoiding the traditional problem of morphological segmentation: *"the a priori marker selection"*. In fact, in a coding context, a good segmentation is the one that gives the best visual quality at the lowest coding cost; it is not easy to derive from this criterion a set of markers for generic images. This is why we have investigated a method which, instead of starting the merging process from a set of markers, allows to merge any two adjacent regions (as proposed in [**?**]). The merging criteria may then take into account the representation and coding power of the subsequent stages of the codec: two adjacent regions which may be represented by a unique texture model should be merged. Coding them as separate entities would unnecessarily increase the coding cost.

3.3. Processing in "inter"-mode

In video coding, a good time stability in the segmentation is the key for obtaining high compression rates. The reason is that a coherent time segmentation can be correctly predicted by means of motion compensation whereas any new object in the scene has to be fully described and coded. In order to obtain time stability we use two successive frames as in [?]; the first frame contains the previous segmentation, the second the current frame. In [?] previous regions are extended into the current time (step called "projection") and afterwards markers of new regions are extracted from the residue and added to the segmentation. The advantage of this approach is the time stability imposed by the projection step (a region growing algorithm taking as markers the previous segmented regions). On the other hand, new regions require a marker extraction from the residue, that is a complex problem in a general case. The region merging algorithm avoided this problem. It produces a segmentation without artificially imposing a set of markers. We combine the advantages of both methods in the following way: we introduce in the region merging algorithm the notion of marker, signaling an important region. Thus, all marked regions will be present in the final segmentation. In contrast with a region growing algorithm, the region merging algorithm allows the presence of not marked regions in the segmentation. In other words, in a region merging algorithm with markers, any two adjacent regions can merge, except if both are markers. Thus, regions that are non marked may crystalize independently from markers.

In order to segment an image sequence we use a region merging algorithm taking as markers the regions of the previous segmentation. The results of this procedure are the following:

- regions of frame $t-1$ are "projected" into frame t (by fusions of regions $t-1$ with regions t).
- new regions may spontaneously appear by independent crystallization in the current time (fusions of regions t), without an external marker selection.

Table I summarizes the behavior of region growing and region merging algorithms.

	Region Growing	Region Merging
Marker with Non-Marker	Allowed Fusion	Allowed Fusion
Non-Marker with Non-Marker	Forbidden Fusion	Allowed Fusion
Marker with Marker	Forbidden Fusion	Forbidden Fusion

TABLE I
Behavior of Region Growing Algorithm and Region Merging Algorithm With Markers

We have described the region merging algorithm and its framework. Now we will present the implemented merging criteria.

4. Merging Criteria

4.1. Contrast based merging

The contrast criterion is the simplest one, and it is used at the first stages of simplification of an input image. The edge valuation is the grey level difference between two adjacent regions.

4.2. Texture based merging

The idea is to merge those regions whose union is correctly represented by the texture model used in the codec. In this way the coding cost is reduced whereas visual quality is not significantly affected [**?**]. To do so, we need a measure that valuates the texture resemblance of adjacent regions.

4.2.1. *Texture resemblance measures of adjacent regions*

Resemblance of texture parameters This measure is based on the comparison of the texture parameters themselves. Since the texture parameters we use are dependent on the shape of the regions (i.e. orthogonal polynomials) this measure is not adapted to our purposes.

Loss of quality caused by a fusion When two regions merge, only one set of texture parameters (instead of one set by region) represents the union of them. This fact leads to a loss of quality. If we merge regions with the smallest loss of quality, we obtain a segmentation optimized from the point of view of the quality of its regions. The problem of this measure is that it is based on a relative criterion. The same loss of quality is allowed for regions of initial good quality as well as for those whose initial quality was poor. The result is an image of inhomogeneous quality in which defects are accentuated.

Quality of adjacent regions after fusion This measure is based on the representation quality of regions after fusion. In contrast with the previous measure, it produces images of homogeneous quality, which leads to better visual aspects. This is the criterion we are going to use in the following. The quality estimation after a fusion is not computed on the resulting merged region. Quality is independently estimated in each of both regions and their minimum is considered. Otherwise, small regions would be highly damaged when merging with bigger ones, because their contribution to the error would be negligible.

4.2.2. *Region merging algorithm based on texture*

The exhaustive algorithm that maximizes the final quality for a given compression rate would be:

- find the couple of regions whose merging preserves the best quality.
- merge them.
- reevaluate edges around the new region.
- iterate.

However, the computational cost forbids this solution. It requires the computation of a texture model for each couple of adjacent regions. As a suboptimal (and practical) solution, a non exhaustive search algorithm is implemented. Neighborhood graph edges are valuated with a simple similarity measure (i.e. contrast criterion) and considered one by one in increasing order of this valuation. Before removing an edge (merging two regions), the merging is validated by a quality criterion: if the quality after fusion is satisfactory, the merger is performed; otherwise, both regions remain as two different regions in the final segmentation.

4.3. Motion based merging

The coding of an image in INTER mode consists in compensating in motion the previous coded image and next in coding the prediction error (of contours and texture). Thus, we need a motion vector for each segmented region. But if several regions have a coherent motion (which is frequently the case), they can be correctly compensated together, with the consequent reduction of the coding cost (less contour pixels, less motion vectors and less parameters of texture correction).

To implement this criterion we have to face the same problems than in the previous section. The solutions we have adopted are the following:

- we have to define a motion similarity criterion: motion resemblance valuation is based on the compensation quality after their fusion. Quality is estimated independently on both regions and the min of them is considered.
- due to the fact that the computation of a motion vector for each couple of adjacent regions leads to an expensive algorithm in terms of computational time, we have implemented a suboptimal algorithm. Instead of calculating a motion vector for a couple of regions, we consider $\vec{v}_1$ (motion vector of R_1) and $\vec{v}_2$ (motion vector of R_2) as two approximations of $\vec{v}$ (motion vector of $R_1 \cup R_2$) and we choose between them the one that leads to a better compensation quality. Figure 2 illustrates this procedure.

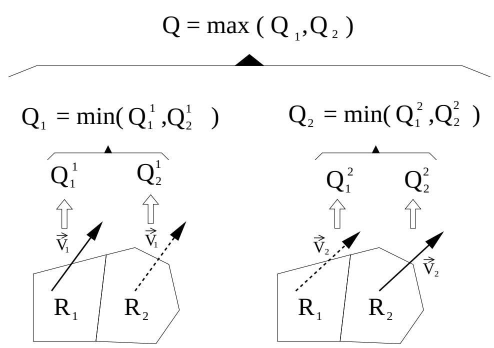


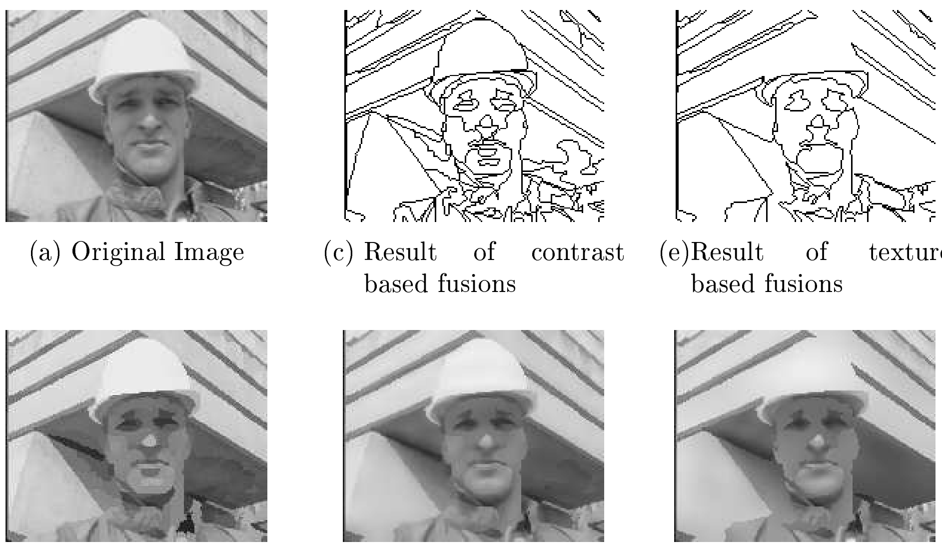
Fig. 2. Suboptimal computation of the compensation quality after fusion.

5. Results

Simulations have been performed using Foreman test sequence in QCIF format (176 x 144).

Fig. **??** shows intermediate stages of the intra segmentation. Fig **??**(a) contains the original image and fig. **??**(b) the oversegmentation, with 735 regions, obtained after the pre-processing stage. We apply to this oversegmentation a region merging algorithm based on contrast and the resulting segmentation (fig. **??**(c)) contains 99 regions. This image is simplified by a texture based merging: regions whose quality after fusion are over 29 dB are merged. The resulting segmented image contains only 52 regions while the quality of the coded image is not significantly affected.

An example of the motion based region merging algorithm is presented in figure **??**. Figures **??**(a) and (b) show the two original images. Using backward motion estimation of [**?**] with the segmentation of figure **??**(c) (92 regions) we obtain the image of figure **??**(d). The quality of this compensation is 28.5 dB. Merging those regions that can be compensated together we obtain the segmentation of the figure **??**(e) (35 regions). The resulting compensated image is presented in figure **??**(f). Its quality is 26.7 dB.



(a) Original Image

(c) Result of contrast based fusions

(e)Result of texture based fusions

(b) Pre-processed Image

(d) Coded image using segmentation(c)

(f)Coded image using segmentation(e)

Fig. 3. Texture simplification

6. Conclusion

A region merging algorithm has been implemented that gives a great flexibility to introduce complex criteria in the segmentation stage. We have presented criteria that take into account coding information in order to improve the efficiency of the codec.

(a) Original Image 1

(c) Segmentation of image (b)

(e) Motion simplified segmentation

(b) Original Image 2

(d) Compensated Image using segmentation (c)

(f) Compensated Image using segmentation (e)

Fig. 4. Motion simplification.

For the segmentation of a real image we use a combination of these criteria. Fusion criteria are used one after the other in increasing order of complexity: contrast-based merging produces a first segmentation that is simplified by texture and motion criteria.

Finally, we have combined the advantages of a marker approach with the region merging algorithm in order to produce stable segmentations in time for images sequences.

References

1. LEP. First results of motion prediction based on a differential estimation method. In *SIM(94) 35 COST 211*, Tampere, June 1994.
2. B. Marcotegui, J. Crespo, and F. Meyer. Morphological segmentation using texture and coding cost. *IEEE Workshop on Non Linear Signal/Image Processing*, Juin 1995.
3. F. Meyer. Algorithmes à base de files d'attente hiérarchique. Technical Report NT-46/90/MM, Ecole des Mines de Paris, Centre de Morphologie Mathématique, September 1990.
4. O. Monga. *Segmentation d'Images par Croissance Hiérarchique de Régions*. PhD thesis, Université Paris Sud. Centre d'Orsay, 1988.
5. M. Pardàs and P. Salembier. Time-recursive segmentation of image sequences. In *EUSIPCO-94*, Edinburgh, Sept 1994.
6. J. Serra and P. Salembier. Connected operators and pyramids. In *Proceedings of SPIE, San Diego*, 1993.
7. L. Vincent. Greyscale area openings and closings, their efficient implementation and applications. In *Workshop on Mathematical Morphology, Barcelona*, pages 22–27, May 1993.