REGION BASED ANALYSIS OF VIDEO SEQUENCES WITH A GENERAL MERGING ALGORITHM*

Luis Garrido and Philippe Salembier
Universitat Politècnica de Catalunya, Campus Nord – Módulo D5
C/ Gran Capità, Barcelona 08034, Spain
Tel: (+34) 93 401 1627, Fax: (+34) 93 401 6447
E-mail: {oster, philippe}@gps.tsc.upc.es
http://gps-tsc.upc.es/imatge

ABSTRACT

Connected operators [4] and *Region Growing* [2] algorithms have been created in different context and applications. However, they all are based on the same fundamental merging process. This paper discusses the basic issues of the merging algorithm and presents different applications ranging from simple frame segmentation to video sequence analysis.

1 INTRODUCTION

Several merging techniques have been developed for image segmentation and filtering purposes. Filters are generally used for preprocessing to remove part of the image content such as noise or irrelevant details. For a segmentation application these filters should simplify the image preserving the shape of remaining objects. Filters possessing this property are known as *connected operators*. They select some objects of the image and remove them by merging the associated zones where the signal is constant (usually called flat zones). Therefore *connected operators* do not introduce any new contour in the image. A large number of segmentation tools are also based on merging techniques. The classical *Region Growing* algorithm is based on merging initial regions with individual neighboring pixels (belonging to an uncertainty area) based on a similarity measure between them.

The goal of this paper is to define a general merging algorithm (section 2) suitable for both (image and video) segmentation and filtering purposes. An efficient implementation of the algorithm will be needed (section 3) in order to be able to use it for sequence segmentation purposes. Our main interest is to develop a set of tools for automatic video segmentation and analysis (section 4) oriented towards the *MPEG-4* and *MPEG-7* standard.

2 GENERAL MERGING ALGORITHM

The proposed algorithm works on a *Region Adjacency Graph* (RAG). The RAG is a set of nodes representing regions (connected components) of the space and a set of links connecting every two neighboring nodes. Note that a node of the graph can represent either a region, a flat zone or even a single pixel. A merging algorithm on this graph is simply a technique that removes some of its links and merges the corresponding nodes. We will assume that the merging is done in an iterative way.

Let R_1 and R_2 be two neighboring regions. In order to completely specify a merging algorithm one has to specify three notions:

1. The *merging order*: it defines the order in which the links are processed. This order, $\mathcal{O}(R_1, R_2)$, is a real value and is a function of the neighboring regions R_1 and R_2.

2. The *merging criterion*: each time a link is processed, the merging criterion decides if the merging has to be done or not. It is also a function of the neighboring regions R_1 and R_2, $\mathcal{C}(R_1, R_2)$, but it can only take two values: *merge* or *do not merge*.

3. The *region model*: when two regions are merged, the model defines how to represent the union. Let us denote this model by $\mathcal{M}(R)$.

In the case of a *Region Growing* algorithm, the *merging order* is defined by the similarity measure between two regions, the *merging criterion* states that all the mergings have to be done until a termination criterion is reached, and the *region model* usually used is the mean.

For classical connected operators [4] the *merging order* is given by the maximum gray level of R_1 and R_2, the *region model* is the minimum gray level of both regions, and the *merging criterion* acts as a sieving tool [3].

Note that the *merging order* is related to the similarity measure of the *Region Growing* algorithm and that the selection characteristic of the *Connected Operators* is introduced by the *merging criterion*.

With the proposed merging algorithm not only classical merging techniques can be implemented but also new merging tools can be developed taking advantage of both filtering and segmentation viewpoint.

*This work has been supported by France-Telecom/CCETT under the contract 96ME22

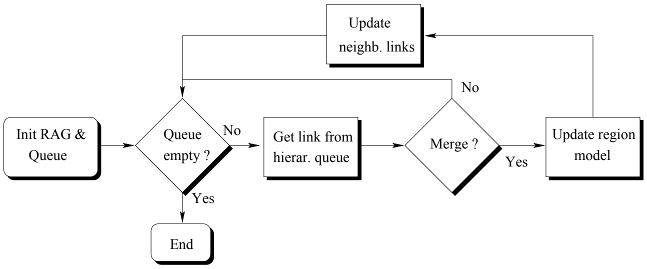

Figure 1: Block diagram of the merging algorithm.

3 EFFICIENT IMPLEMENTATION

3.1 Merging Algorithm

In fig. 1, the general scheme of the merging process is represented. The merging algorithm can be divided into two stages.

The first one is devoted to the initialization of the structures needed for the merging, i.e. the RAG structure and a hierarchical queue. Each region is initialized by computing its *model*. Then the *merging order* for each pair of neighboring regions is computed. This order (represented a floating point number) is used to insert the links of the graph into a hierarchical queue (i.e. a queue in which elements with higher priority are served first).

The second stage consists in the merging procedure itself. The algorithm begins to extract from the hierarchical queue the link with highest priority. Then the *merging criterion* decides if the merging has to be done or not. If the algorithm decides not to merge the regions, the algorithm returns to the first block of the merging algorithm. Note that this decision is final in the sense that the link will never be removed since it is not reintroduced in the queue. If the merging criterion decides to remove the link, the new *model* for the merged region is updated and the *order* of the neighboring links is recomputed. The latter implies the extraction of the corresponding links of the queue, the computation of the new *merging order* and the insertion of these links into the queue with their new priority. At this point the merged RAG structure has been computed and updated: the iterative process starts again by checking if the queue is empty.

One of the main keys of the efficient implementation of the merging algorithm is the hierarchical queue. For more information on this subject please see [3].

3.2 Efficiency of the Algorithm

The efficiency of the algorithm turns out to be very high. Suppose that the merging algorithm is started from the pixel level (i.e. each node is associated to one pixel) and ended once the partition is made of one single region. If a simple model is used (constant within the region), the *CPU* time needed is about 1.5 seconds on a *Pentium 200MHz* for a QCIF gray-level image.

4 REGION BASED ANALYSIS

Different applications ranging from intra frame segmentation to video sequence segmentation and analysis have been implemented with the merging algorithm introduced in the previous section. Some results for region based analysis will be shown.

4.1 Intra Frame Segmentation

Our goal is to segment an image by defining homogeneous regions in color. For that purpose, and for simplicity, the case of gray-level segmentation will be explained first.

The *model* $\mathcal{M}(R)$ that will be used is the mean gray level of the pixels belonging to the region. That is, a zero order model is used. The *merging order* should give us a measure of the likelihood that two regions belong to the same object. From our practical experience, in order to obtain a good compromise between region detection and contour accuracy, a simple but good merging order is [3]

$$\begin{aligned} \mathcal{O}(R_1, R_2) &= N_1[\mathcal{M}(R_1) - \mathcal{M}(R_1 \cup R_2)]^2 \\ &+ N_2[\mathcal{M}(R_2) - \mathcal{M}(R_1 \cup R_2)]^2 \end{aligned}$$

where N_1 and N_2 denote the number of pixels of R_1 and R_2 respectively.

In the case of multichannel images, such as a color image, a model is assigned to each component of the image, and the order is defined to be a linear combination of the order values defined on each component.

The *merging criterion* simply defines the end of the merging process. Two useful termination criteria are the number of regions and the Peak Signal to Noise Ratio between the original image and the modeled one.

Fig. 2 shows the result obtained by segmenting a color image in the YUV space. The merging algorithm has started from the pixel level and ended once a *PSNR* of $27dB$ has been reached. The segmented image is made up of 65 regions.

a) Original image b) Segmented image

Figure 2: Intra frame color segmentation.

4.2 Motion Segmentation

The same strategy can be used to deal with motion oriented segmentation. Assume that we start from a spatial segmentation (for example the one of fig. 2b). The motion estimation, based on differential methods [1, 5], assigns to each region a polynomial model describing

Original partition (63 reg.)

Motion partition (4 reg.)

Figure 3: Example of motion segmentation.

the apparent motion in the horizontal and vertical directions.

In this case a first order *model*, $\mathcal{M}(R) = \alpha i + \beta j + \gamma$ (i, j denote the spatial coordinates of the pixels), is used. The *merging order* used is based on the computation of the increase of the *DFD* when R_1 and R_2 are merged

$$\mathcal{O}(R_1, R_2) = \min_{K \in \{R_1, R_2\}} \left(\frac{\sum_{x \in R} |I_T(x) - I_{T-1}(x - d_K(x))|}{\left(\sum_{x \in R} |\nabla I_T(x)|\right) / N} \right)$$

where $R = R_1 \cup R_2$, I_T and I_{T-1} are the frames at $t = T$ and $t = T - 1$ resp., N is the number of pixels of R, and $d_K(x)$ is the motion vector at position x associated to the extension of the motion model of region K to the whole image support. Note that the *DFD* is normalized by the gradient of the image in order to reduce the effect of inaccurateness of the motion estimation.

The *merging criterion* states that all the mergings have to be done until a global *DFD* is reached. As a result, the algorithm defines regions that are homogeneous in the sense of motion. The result of the segmentation is shown in fig. 3.

In this case no motion re-estimation is performed during the merging process. This choice has been selected in order to reduce the computational complexity of the algorithm.

4.3 Frame Analysis

An image analysis technique often used in mathematical morphology is the so called granulometry [6]. The approach consists in using a hierarchical filtering structure in order to measure a characteristic at each filter output. This allows the characterization of what has been removed by each filter. Considering the segmentation algorithm as a filtering tool, the same approach can be used to characterize the content of each frame. Assume, for example, that the segmentation utilizes a gray level homogeneity criterion and that the termination criterion is the *PSNR* (as explained in section 4.1). *PSNR* values ranging from $45dB$ to $10dB$ are used. Three frames of the *Foreman* sequence have been processed (fig. 4). These curves, called granulometric curves in granulometric analysis, show us how many regions are necessary to achieve a certain $PSNR$.

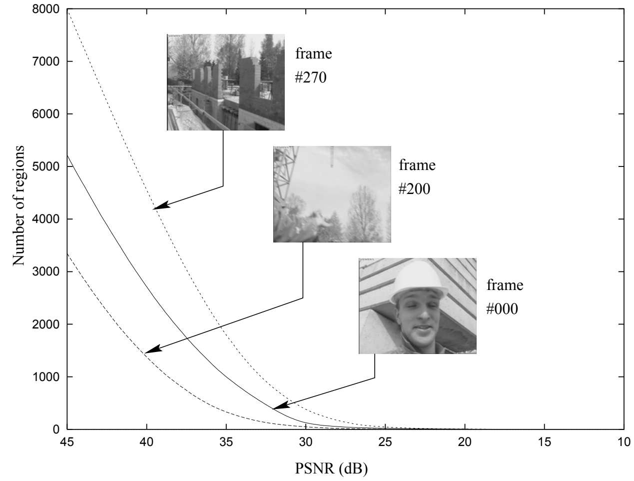

Figure 4: Number of regions versus *PSNR* for different frames of the foreman sequence.

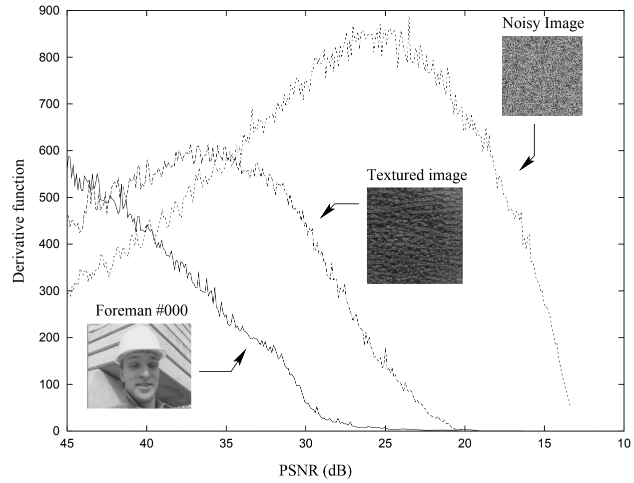

Figure 5: Derivative function of the granulometric curve.

Intuitively, the different number of regions obtained for different frames is due to the image "complexity": for simple frames with few objects (homogeneous in gray level), the number of regions necessary to achieve a given *PSNR* is rather low. Inversely, if the image has a large number of contrasted objects, a high number of regions is necessary to reach the same *PSNR*.

The derivative of the granulometric curve (fig. 4) is also often used in granulometric analysis. In the example of fig. 5 three different images with different types of texture have been processed. The results are shown in fig. 5. Observe that the more the image is granulated, the more the maximum of the derivative moves to the right.

The presented method can be used for video shot detection (each shot should be composed of frames of different complexity) or for complexity-based frame and video indexing.

4.4 Video Sequence Analysis

In this section we focus on a technique for sequence segmentation based on the approach described in section 2 which is discussed in detail in [3]. The goal of this algorithm is to segment a video sequence in a recursive and causal way. To this end, we propose to define at each time t instant a gray partition $P_g(t)$ (partition whose regions are homogeneous in gray-level) and a motion partition $P_m(t)$ (partition whose regions are homogeneous in motion). The gray level partition is created by merging regions belonging to a (very) fine partition such as the partition of flat zones $P_{fz}(t)$. Partition $P_m(t)$ is created by merging regions belonging to $P_g(t)$. Moreover, region tracking is performed for P_g and P_m. That is, regions of $P_g(t)$ ($P_m(t)$) are related to regions of $P_g(t-1)$ ($P_m(t-1)$).

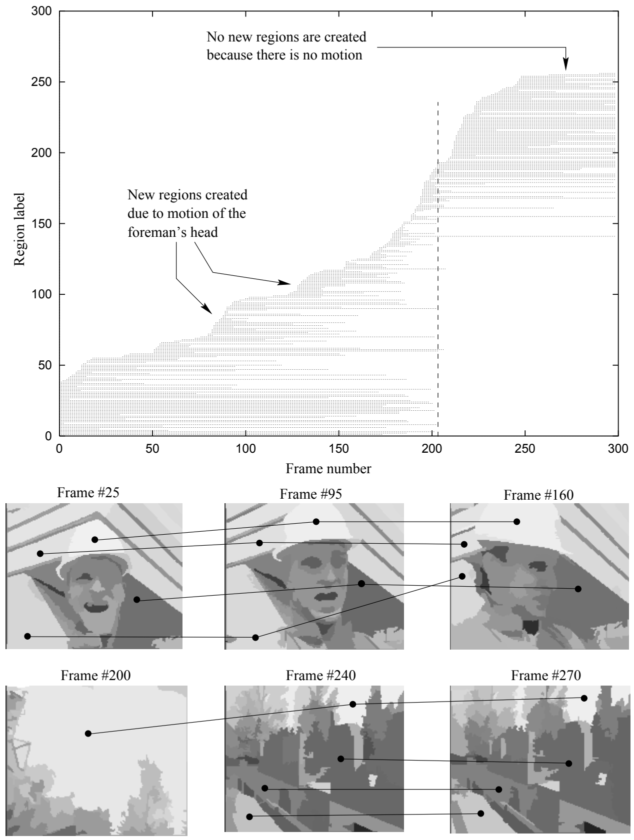

Figure 6: Top: life span of the regions of the gray level homogeneity partition for the Foreman sequence. Bottom: several frames of the foreman sequence with some links established by the tracking algorithm.

Tracking is useful for sequence analysis. One of the main applications is to study the evolution of the regions that compose the partition. In fig. 6 the life span of the regions that composes the gray level partition is plotted. That is, for each frame the set of region labels associated to the regions of the gray level partition is plotted. By inspecting this plot we are able to see when new regions appear as well as when other disappear. Below the plot several frames of the sequence have been drawn superimposed with some temporal links created by the algorithm. This shows us that the tracking is done quite well.

The life span plot can be used to locate frames which involve much (or little) motion by identifying sections of the sequence where a lot of (or very few) new regions are created. Observe, for example, that at frames $90-100$ and $140-150$ a lot of new regions are created due to the motion of the foreman's head. Similarly, at frames $250-300$ no new regions are created since there is no motion in these frames.

Another application is the splitting of the sequence in several parts according to the general behavior of the life of regions. In the example of fig. 6 two sections can be identified. The first one involves frames 0 up to 200 while the second one involves frames 200 to 300. Notice that the panning of the camera around frames 200 leads to the disappearance of all the regions that were involved in the partition in the first part of the sequence.

5 CONCLUSIONS

A general merging algorithm has been presented. It is based on three notions: *merging order*, *merging model* and *merging criterion*. The merging algorithm allows the implementation of classical region based segmentation and connected operators, and also new region based merging techniques. We have shown several applications ranging from simple intra frame segmentation using different criteria (gray level and motion) to a more complex scheme dealing with sequence segmentation.

References

[1] J.L. Dugelay and H. Sanson. Differential methods for the identification of 2D and 3D motion models in image sequences. *Signal Processing, Image Communication*, 7:105–127, 1995.

[2] A.K. Jain. *Fundamentals of Digital Image Processing*. Prentice-Hall, 1989.

[3] P. Salembier, L. Garrido, and D. García. Image sequence analysis and merging algorithms (invited paper). In *Int. Workshop on Very Low Bit-Rate Video*, pages 1–8, Linkoping, Sweden, july 1997.

[4] P. Salembier and J. Serra. Flat zones filtering, connected operators and filters by reconstruction. *IEEE Transactions on Image Processing*, 7:1153–1160, August 1995.

[5] H. Sanson. Toward a robust parametric identification of motion on region of arbitrary shape by non-linear optimization. In *IEEE Int. Conf. on Image Processing*, volume 1, pages 203–206, October 1995.

[6] J. Serra. *Image Analysis and Mathematical Morphology*. Academic Press, 1982.