Morphological Tools for Robust Key-region Extraction and Video Shot Modeling [⋆]

Javier Ruiz Hidalgo and Philippe Salembier

Technical University of Catalonia, Barcelona, Spain.
Email: {jrh,philippe}@gps.tsc.upc.es

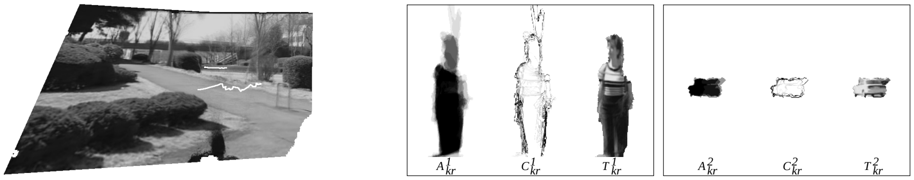
Abstract. In recent years, the use of multimedia content has experienced an exponential growth. In this context, the need of new image/video sequence representation is becoming a necessity for many applications. This paper deals with the structuring of video shots in terms of various foreground key-regions and a background mosaic. Each key-region represents different foreground objects that appear through the entire sequence in a similar manner the mosaic image represents the background information of the complete sequence. We focus on the interest of morphological tools such as connected operators or watersheds to perform the shot analysis and the computation of the key-regions and the mosaic. It will be shown that morphological tools are particularly attractive to improve the robustness of the various steps of the algorithm.

1 Introduction

Images and video sequences modeling is experiencing important developments. Part of this evolution is due to the need to support a large number of new multimedia services. Traditionally, digital images were represented as rectangular arrays of pixels and digital video was seen as a flow of frames. New multimedia applications can rely on indexing or content-based coding that allow a representation that is more structured and hopefully closer to the real word.

The most straightforward way of representing video shots is to consider them as a set of contiguous frames. An alternative approach is to represent them by a subset of representative frames called key-frames. A more sophisticated approach for shot representation involves the analysis of the spatio-temporal content of the video shot. In [5] and [7], for instance, the representation of a video shot is composed of a set of layers representing the background information and various foreground layers. An attractive background representation relies on mosaic images [5,1]. Mosaics are panoramic views of the background components that are visible during the shot [5,7]. Mobile foreground objects can then be superimposed to the mosaic representation. In the sequel, these foreground objects will be represented by *key-regions*. A typical example of shot representation based on background mosaic and key-regions is shown in Fig. 1. The background mosaic is presented in Fig. 1.a and two key-regions are shown in Fig. 1.b and 1.c. Each key-region is represented here by an appearance image A_{kr}^{k}, a contour image C_{kr}^{k} and a texture image T_{kr}^{k}, where kr stands for key-region and k the key-region number. The meaning and computation of these images will be presented in this

[⋆] The authors would like to thank the support of the European Commission and in particular, the ACTS DICEMAN Project.



a) Background Mosaic b) Key-region 1 (girl) c) Key-region 2 (car)

Fig. 1. Video shot representation with a background mosaic a) and two key-regions b) and c). Key-regions are represented from left to right by an appearance image A_{kr}, a contour image C_{kr} and a texture image T_{kr}.

paper. Note that the motion trajectories of the key-regions are also drawn (as white lines) on the background mosaic.

The extraction of foreground regions in video sequences is an active research topic. Classical approaches [5,7] mainly rely on motion information. However, pure motion-based algorithms fail when shots present rapidly changing backgrounds, when foreground objects present little motion with respect to the camera or when foreground objects have a low contrast with respect to the background. The shot representation technique proposed in this paper builds, in a first step, a background mosaic and then uses this mosaic to extract key-regions. Beside the explanation of the complete algorithm, the main focus of this paper is to highlight the use of morphological tools such as connected operators [4] and watersheds to improve the robustness of the algorithm [2,6].

This paper is organized as follows. Section 2 gives an overview of the proposed algorithm. Section 3 presents the use of motion-oriented connected operators for outliers detection in the mosaic creation algorithm. Section 4 explains the foreground segmentation algorithm and section 5 the creation of key-regions. The representation and modeling of key-regions are discussed in section 6. Finally, conclusions are drawn on section 7.

2 Overview of the algorithm

The algorithm is highlighted in Fig. 2 and involves three steps. The first one is the background mosaic computation (top blocks of Fig. 2). The second step extracts the shape of each key-region at each time instant (middle blocks) and the last step combines the information obtained at each time instant and builds the key-region models (bottom blocks). Next sections will describe each step.

The background mosaic computation follows a classical approach [1]. The first step is to compute the dominant motion between successive input images, $I(t)$ and $I(t-1)$. The dominant motion, $m(t)$, is assumed to represent the camera motion and is used to warp the original frames in the same coordinate system. The warped images are blended to produce the mosaic image, I_{mos}. In order to be robust, the blending should only take into account pixels belonging to the background. As a result, before the warping and blending step, outliers that do not follow the dominant motion are identified. They are represented by an outliers mask $M_{out}(t)$. In section 3, we will show how morphological connected operators efficiently allow the identification of outliers.

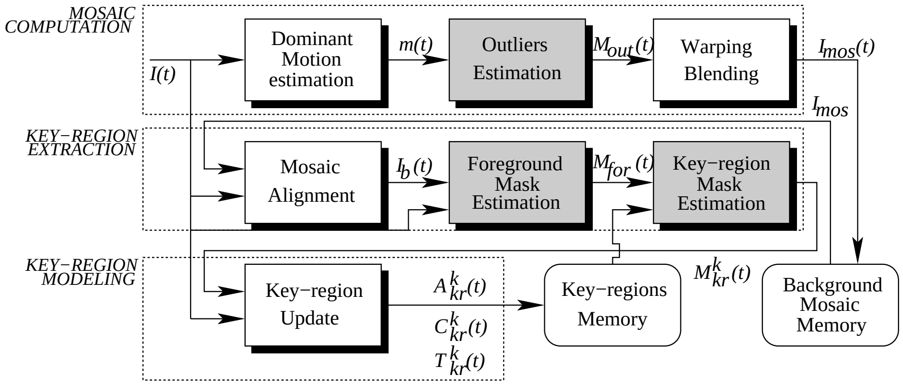


Fig. 2. Overview of the algorithm. Blocks in gray represent steps where morphological tools are used (only the major input and output signals are represented).

The second step extracts, for each key-region k at time t, a key-region mask $M^k_{kr}(t)$. This extraction starts by the mosaic alignment. Its goal is to produce an estimation of the background information, $I_b(t)$, at time t. Taking into account the dominant motion, the relevant part of the mosaic is un-warped to be compared to the current image $I(t)$. A foreground mask, $M_{for}(t)$, is computed by comparing the original image $I(t)$ with the background estimation $I_b(t)$. A watershed algorithm (section 4) is used for this step. The foreground mask $M_{for}(t)$ is an estimation of the key-regions at time t. However, this estimation is not very reliable because it is obtained on the basis of the observation of a single time instant. To improve the robustness of the analysis, the last step combines the contour information of the foreground masks extracted at each past time instant and selects the most reliable sections that have been observed to create the mask of the key-region, $M^k_{kr}(t)$. A watershed algorithm can be also used to combine a set of contours taking into account their reliability (section 5).

Finally, the last step of the algorithm takes into account the key-region masks, $M^k_{kr}(t)$, as well as the original image, $I(t)$, to update the key-region models (see section 6 for more details). In the following sections, we explain the use of morphological tools for the outliers estimation (section 3), the foreground mask estimation (section 4) and the key-region mask estimation (section 5).

3 Outliers estimation with connected operators

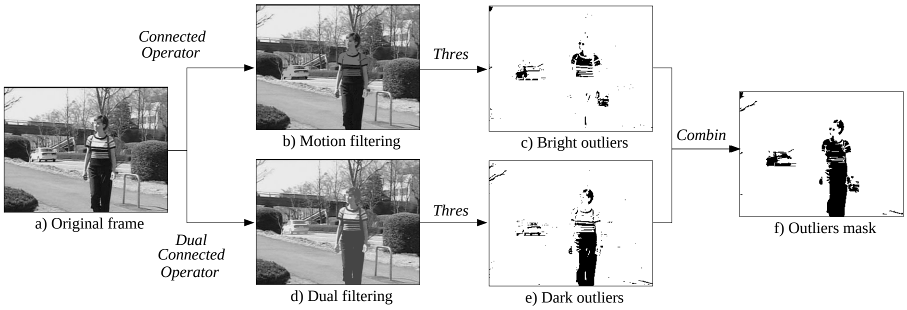
Morphological connected operators are used to detect and remove outliers that do not follow the dominant mosaic motion in the mosaic creation step. Gray level connected operators are operators that act by merging elementary regions called *flat zones* [4]. They cannot create new contours or modify the position of existing boundaries between regions and, therefore, have very good contour preservation properties. Several approaches can be used to create connected operators. We will use the one discussed in [3]. The strategy consists in creating a region-based tree representation of the image and to apply a pruning strategy on the tree to simplify the image (in this case, without the outliers).

The tree representation is called Max-tree and is oriented towards signal maxima. Each node $\mathcal{N}_i$ in the tree represents a connected component of the space that is extracted by the following thresholding process: for a given threshold value T, consider the set of pixels X that have a gray level value larger than T and the set of pixels Y that have a gray level value equal to T:

$$X = \{x, \text{ such that } f(x) \geq T\} \text{ and } Y = \{x, \text{ such that } f(x) = T\} \tag{1}$$

The nodes $\mathcal{N}_i$ represent the connected components of X such that $X \bigcap Y \neq \emptyset$.

The filtering strategy consists in pruning the tree and in reconstructing the image from the resulting pruned tree. The simplification is governed by a criterion which may involve simple notions such as size, contrast or more complex ones such as texture, motion or even semantic criteria. Here, the detection of outliers is based on a motion criterion. For all input frames, the corresponding max-tree is created. A recursive version of the mean displaced frame difference is computed for all nodes of the trees using the dominant mosaic motion $m(t)$ [3]. Nodes of the tree that do not follow the given motion produce a high displaced difference and should be removed. The criterion is not increasing: there is no constraint stating that if a node has to be removed, its children have also to be removed. Therefore, a dynamic programming strategy based on the Viterbi algorithm is used. We refer the reader to [3] for a complete description of the max-tree creation and the morphological filtering involved.



a) Original frame
b) Motion filtering
c) Bright outliers
d) Dual filtering
e) Dark outliers
f) Outliers mask

Fig. 3. Estimation of outliers with connected operators

Fig. 3 shows an example of outliers estimation. Connected operators remove maxima of the image that do not follow the dominant mosaic motion. Fig. 3.b and 3.c show an original frame and the output of the connected filter. The filter has removed the bright components of the outliers (the girl and the car) and has preserved the background information. Comparison between the original and filtered frames gives the mask corresponding to bright outliers. The estimation of dark outliers can be done using the dual connected operator. The dual operator ψ^* is defined by: $\psi^*(f) = -\psi(-f)$ and has the same effects as ψ but on minima. Fig. 3.e and 3.f show the filtered output and the mask corresponding to dark outliers. The final outliers mask is shown in Fig. 3.g.

On the other hand, classical mosaic creation algorithms try to remove outliers by defining a map assigning to each pixel a value representing whether it belongs to the foreground or to the background. The classical value assigned to each pixel of the weight map image is:

$$w(t)[x] = \frac{c}{c + |I(t)[x] - I(t-1)[x - m(t)[x]]|^2} \tag{2}$$

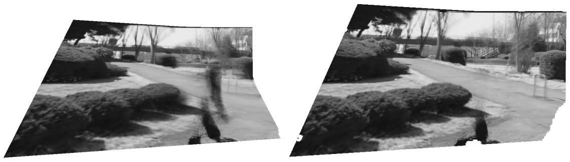


Fig. 4. Comparison of mosaic creation without a) and with b) connected operators.

Fig. 4 compares the classical solution with the one proposed using connected operators. The classical approach does not allow the elimination of outliers that occupy a significant portion of the image (as the girl). A dark shadow is clearly visible in the lower right part of the mosaic of Fig. 4. Moreover, the partial elimination of outliers has a strong effect on the successive warping and blending steps: strong geometrical deformations appear on the lower right part of Fig. 4.a.

4 Foreground mask estimation with watershed

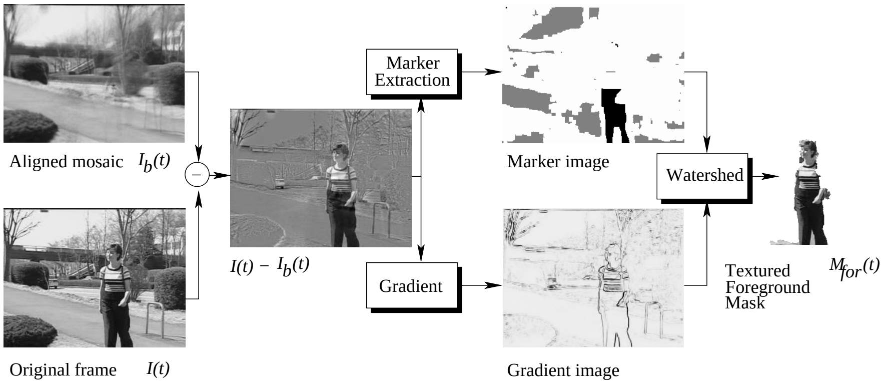


Fig. 5. Estimation of the foreground mask

The foreground mask extraction process is outlined in Fig. 5. Using the dominant motion, the relevant part of the mosaic is un-warped to produce an estimation of the background $I_b(t)$ at time t, that can be compared to the current image $I(t)$. All the relevant information is concentrated in the image: $I(t) - I_b(t)$. The foreground mask, $M_{for}(t)$, is computed by using a watershed algorithm [6]. The watershed algorithm is applied on a gradient image and uses markers to initiate the propagation process.

The gradient image should indicate the contours of the foreground mask. It is mainly computed from the image gradient of $I(t) - I_b(t)$. However, this gradient highlights contours but also textured areas. To solve this drawback, the gradient is weighted (pixel by pixel) by a temporal gradient: $\mathcal{G}\{I(t) - I(t-1)\}$, where $\mathcal{G}$ denotes the gradient operator:

$$G(t) = \mathcal{G}\{I(t) - I_b(t)\} \cdot (\mathcal{G}\{I(t) - I(t-1)\} \vee \mathcal{G}_0) \tag{3}$$

where $\vee$ denotes the maximum and $\mathcal{G}_0$ is used as lower-bound of the weighting gradient so that the weight is not too low on static areas.

Markers are obtained by thresholding $|I(t) - I_b(t)|$ and by erosion of the resulting masks. Two different thresholds, t_{for} and t_{back}, are used to extract foreground and background makers. Assume that $\epsilon_s\{\cdot\}$ denotes a binary erosion with a structuring element, s. The foreground and background markers are defined by $M_{for}(t) = \epsilon_s\{|I(t) - I_b(t)| > t_{for}\}$ and $M_{back}(t) = \epsilon_s\{|I(t) - I_b(t)| < t_{back}\}$ respectively. The threshold values were empirically chosen to be $t_{for} = 35$, $t_{back} = 10$ and s an square structuring element whose length is 2 per cent of the original image size. Results have shown these values to be very robust even across different type of sequences. Foreground and background markers are combined in a single image called *Marker image* in Fig. 5. In this image, the dark (grey) areas correspond to foreground (background) markers.

Finally, the watershed is applied to the gradient image $G(t)$ using the markers, $M_{for}(t)$ and $M_{back}(t)$. A final step groups all connected regions into the same connected masks and considers non-connected regions as different foreground regions. The segmentation can be seen on the right side of Fig. 5 where the girl has been successfully segmented from the background.

5 Key-region mask definition with watershed

The foreground mask $M_{for}(t)$ is an estimation of the key-regions at time t. However, this estimation is not very reliable because it is obtained on the basis of the observation of a single time instant. To improve the robustness of the analysis, the key-region mask estimation step combines the contour information of the foreground masks extracted at past time instants and selects the most reliable sections to create the mask of the key-region k, $M_{kr}^{k}(t)$.

The first step of the algorithm is to associate connected components of the background mask $M_{for}(t)$ to key-regions that are already stored in the key-region memory. A connected component of the foreground mask is assigned to an existing key-region if it sufficiently overlaps with the last assigned foreground mask of the corresponding key-region. This approach works well on common scenes where changes between frames (at 25 or 30 fps) are usually small. If the current foreground mask does not correspond to any known key-region, a new key-region is created.

Once a connected component of the foreground mask is assigned to an existing key-region, it should be aligned to the same coordinate system. This alignment is performed by estimating the motion between the foreground mask and the stored key-region. After alignment, let us denote by $\hat{M}_{for}(t)$ and $\hat{I}(t)$ the motion compensated version of the foreground mask and the motion compensated input image. These images can be seen on the left side of Fig. 6. Note that, in this example, the contour of the foreground mask is not always reliable.

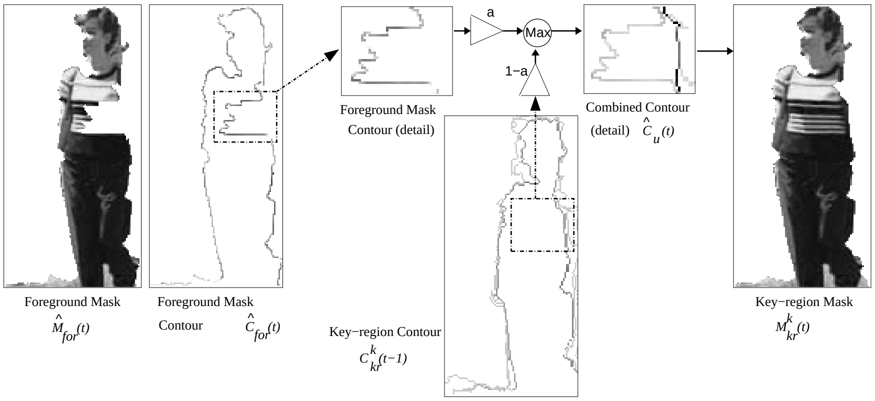


Fig. 6. Creation of the key-region mask $M^k_{kr}(t)$ taking into account the reliability of the current foreground mask $\hat{M}_{for}(t)$ and of past contour information $C^k_{kr}(t-1)$.

Our goal is to combine this contour information with the contour information of the same key-region extracted at previous time instants taking into account the reliability of contours in time. The update of the foreground shape starts from the compensated foreground mask $\hat{M}_{for}(t)$ and is performed as follows. Assume that I is an image and M a mask, $\mathcal{C}\{I, M\}$ denotes an image equal to zero except on the contours of M where it takes the values of I. The contour reliability of the foreground mask $\hat{M}_{for}(t)$ is obtained by:

$$\hat{C}_{for}(t) = \mathcal{C}\left\{\mathcal{G}\{\hat{I}(t)\}, \hat{M}_{for}(t)\right\} \tag{4}$$

The pixels value of this contour image is a confidence measure of the contours of the current foreground mask $\hat{M}_{for}(t)$. Low values imply that the corresponding contour does not correspond to contrasted edges. This can occur, for instance, when the foreground occludes a background region of the same color. High values on the contour measure correspond to strong edges on the original image and therefore to reliable contours. Fig. 6 illustrates the use of the contour image to correct possible segmentation errors in the foreground mask extraction algorithm. In this example, the foreground mask extracted at frame 2039 of the *nhkvideo7* sequence of the *MPEG-7* database is of poor quality due to a low contrast between the girl and the background in that specific time instant.

The two images on the left side of Fig. 6 show the extracted foreground mask $\hat{M}_{for}(t)$ and the measure of contours reliability $\hat{C}_{for}(t)$ (as in Equ. (4)). This measure of the contour confidence on the current foreground mask is compared with the same accumulated measures from the previous foreground masks assigned to the same key-region, $C^k_{kr}(t-1)$. This accumulated contour image is part of the key-region model (see section 6). The corresponding accumulated contours measure of the key-region is shown on the bottom image of Fig. 6. The combination of the current contour $\hat{C}_{for}(t)$ and the accumulated contours $C^k_{kr}(t-1)$ is done by a maximum operation:

$$\hat{C}_u(t) = a\hat{C}_{for}(t) \vee (1-a)C^k_{kr}(t-1) \tag{5}$$

The parameter $a \in [0,1]$ controls the memory of the allowed modifications to the shape of extracted foreground masks. If $a \simeq 0$, previously segmented key-region contours are trusted more than the current contours from the foreground mask. In this case, errors in the foreground mask are easier to fix but tracking non-rigid foreground regions becomes more difficult. On the other hand, if $a \simeq 1$, non-rigid regions are easier to track but segmenting errors are also more difficult to correct. In our case, a value of $a = 0.5$ has been used for all examples. Note that resulting contour values of $\hat{C}_u(t)$ are only used locally as the gradient image for the watershed of the key-region mask definition (see Figure 6) so the implied lowered of the gradient when using $a < 1$ is not propagated on following frames.

The estimation of the final mask of the foreground region: $M^k_{kr}(t)$ is done with a watershed algorithm. The markers for this watershed consist of two points, one inside the foreground mask and one outside (in the background). The output of the watershed algorithm is the new foreground mask $M^k_{kr}(t)$ where the most reliable contour parts from the foreground mask and from the assigned key-region have been used. The resulting mask is shown on the right side of Fig. 6. The initial error in the foreground mask shape has been eliminated and replaced by the most reliable contour observed in the past. In general, this procedure allows the progressive improvement of the key-region contours on a frame by frame basis taking into account the reliability of past extracted key-region contours.

6 Key-Region Modeling

The final step of the algorithm creates and updates a model for each key-region observed in the scene (Key-region update block of Fig. 2). The key-region model consists of a template of three images. An appearance image, a contours image and a texture image. The appearance image $A^k_{kr}(t)$ shows the frequency with which a pixel has been estimated as belonging to key-region k. The contour image $C^k_{kr}(t)$ stores the confidence of the key-region contours and is used to modify the input foreground masks in a frame basis as seen in the previous section. Finally, the texture image $T^k_{kr}(t)$ represents the overall texture of the key-region.

If the key-region mask $M^k_{kr}(t) = 1$ denotes pixels that have been extracted and assigned to key-region k at time t and, $\hat{C}_{for}(t)$ is the contour confidence of the extracted foreground mask (as in Equ. (4)). The equations that update each template image are (pointwise operations are implied):

$$T^k_{kr}(t) = \frac{A^k_{kr}(t-1)T^k_{kr}(t-1) + M^k_{kr}(t)\hat{I}(t)}{A^k_{kr}(t)} \qquad C^k_{kr}(t) = \frac{A^k_{kr}(t-1)C^k_{kr}(t-1) + \hat{C}_{for}(t)}{A^k_{kr}(t)}$$

$$A^k_{kr}(t) = A^k_{kr}(t-1) + M^k_{kr}(t) \tag{6}$$

Fig. 7 shows the key-region template images from a scene where a person walks in front of the camera. The appearance, contour and texture template images contain information of the activity followed by the key-region. In this case, higher body parts (body, chest) show no relative movement while lower parts (legs) show a considerable amount of relative motion. This representation is particularly attractive to analyze the activity of non-rigid regions.

Fig. 1 shows a complete shot representation of the *nhkvideo7* sequence. The background information is separated from the key-regions of the scene. In the

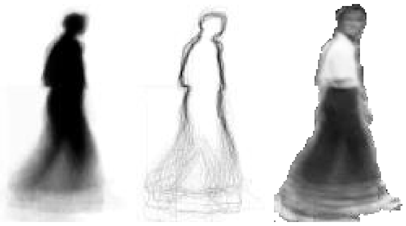


Fig. 7. Modeling of key-region k with, from left to right, an appearance image $A^k_{kr}(t)$, a contour image $C^k_{kr}(t)$ and a texture image $T^k_{kr}(t)$.

original sequence, the camera follows the walking girl while a car crosses the road in the background. Two key-regions have been extracted corresponding to the girl and the car. Fig. 1.b and 1.c show the corresponding template images of the two key-regions. Superimposed to the final mosaic image, the relative motion respect to the camera is drawn.

7 Conclusion

A method for representing and structuring video shots has been presented. A robust outliers detection algorithm based on connected operators is used to estimate and create a mosaic image of the background information of the scene. This background information can then be used to extract representative foreground key-regions that appear in the shot. The proposed approach uses a watershed algorithm to extract the foreground mask on a frame by frame basis. These foreground regions are refined using the reliability of previous extracted contours and are progressively combined into key-region templates. At this step, the watershed algorithm turned out to be again an attractive solution. Both key-regions templates and mosaic image create a compact and useful representation of the content and of the activity of the scene allowing the possibility of further representation, indexing and analysis of the shot.

References

1. J. Davis. Mosaics of scenes with moving objects. In *Proceedings of Computer Vision and Pattern Recognition*, pages 354–360, Santa Barbara, USA, June 1998.
2. F. Meyer and S. Beucher. Morphological segmentation. *Journal of Visual Computing and Image Representation*, 1(1):21–45, 1990.
3. P. Salembier, A. Oliveras, and L. Garrido. Antiextensive connected operators for image and sequence processing. *IEEE Trans. on IP*, 7(4):555–570, April 1998.
4. P. Salembier and J. Serra. Flat zones filtering, connected operators and filters by reconstruction. *IEEE Trans. on IP*, 3(8):1153–1160, August 1995.
5. H.S. Sawhney and S. Ayer. Compact representation of videos through dominant multiple motion estimation. *IEEE Trans. on PAMI*, 18(8):814–830, 1996.
6. L. Vincent and P. Soille. Watersheds in digital spaces: an efficient algorithm based on immersion simulations. *IEEE Trans. on PAMI*, 13:583–598, 1991.
7. J. Wang and E. Adelson. Representing moving images with layers. *IEEE Trans. on IP*, 3(5):625–638, September 1994.