

An Accurate Semi-automatic Segmentation Scheme Based on Watershed and Change Detection Mask

Cédric De Roover^a, Moncef Gabbouj^b and Benoit Macq^a

^aUniversité catholique de Louvain (UCL), Communications and Remote Sensing Laboratory,
Bâtiment Stevin-Place du Levant 2,
B-1348 Louvain-la-Neuve, Belgium

^bTampere University of Technology (TUT), Institute of Signal Processing,
P.O. Box 553, FIN-33101
Tampere, Finland

ABSTRACT

This paper presents a region-based segmentation method extracting automatically moving objects from video sequences. Non-moving objects can also be segmented by using a graphical user interface. The segmentation scheme is inspired from existing methods based on the watershed algorithm. The over-segmented regions resulting from the watershed are first organized in a binary partition tree according to a similarity criterion. This tree aims to determine the fusion order. Every region is then fused with the most similar neighbour according to a spatio-temporal criterion regarding the region colors and the temporal colors continuity. The fusion can be stopped either by fixing a priori the final number of regions, or by markers given through the graphical user interface. Markers are also used to assign a class to non-moving objects. Classification of moving objects is automatically obtained by computing the Change Detection Mask. To get a better accuracy on the contours of the segmented objects, we perform a simple post-processing filter to refine the edges between different video object planes.

Keywords: Segmentation, Watershed, BPT, CDM, fusion

1. INTRODUCTION

In the recent years, a great deal of effort has been devoted to image and video segmentation. Motivations for such a research topic are related to the targeted application. In medical imaging, segmentation can help to detect malignant structures, to evaluate a volume, to count objects within an image. In video surveillance, segmentation can help to detect special events or to track objects over time. Video segmentation and the repartition of a video sequence in different video object planes (VOPs) is also deeply studied for compression purpose. New video coding standards, such as MPEG-4, and video content descriptors, like MPEG-7, are now content-based and required a segmentation process.^{1,2} Another application of object segmentation is the increasing use of augmented reality. In augmented reality, a scene is built from merging virtual and segmented real objects. Segmentation for compression or video surveillance often requires to be unsupervised.^{3,4} Augmented reality applications can be unsupervised, for real-time operations, or supervised for post-processing operations.⁵ Segmentation in medical imaging and video post-production processing are generally supervised to guarantee a control of the resulting accuracy.

Segmentation algorithms can be divide into two main groups : the "region-based" algorithms, that perform the segmentation directly on the pixels, in a discrete domain, and the "active-contours" algorithms, that perform the segmentation in the continuous domain.

The extent of possible applications explains the abundant literature in this domain and why the general problem of segmenting object has not been completely solved yet.

Cédric De Roover is funded by SEGMORA, a project of the Region Wallonne.
E-mail: deroover@tele.ucl.ac.be

In this paper, we investigate a "region-based" segmentation method automatically extracting moving objects from video sequences and semi-automatically extracting static objects by using a graphical user interface. Our approach proceeds by establishing and fusing a set of little homogenous regions in order to classify the image in different object planes.

The segmentation scheme is inspired from existing methods^{6,7} based on the watershed algorithm. The main drawback of the watershed is the resulting over-segmentation (the region to segment is split into numerous sub-regions). To solve this over-segmentation problem we fuse the smallest regions according to a spatio-temporal similarity criterion regarding the region colors and the temporal colors continuity. A binary partition tree indicates the fusion order. The fusion is stopped either by fixing previously the final number of regions, or by markers given through the graphical user interface. These markers are also use to assign a class to non-moving objects. Classification of moving objects is automatically obtained by computing the Change Detection Mask (CDM).

Another drawback of the watershed based algorithms is the poor accuracy of the contours of the segmented objects. This can be explained by the three main following reasons. Firstly, pre-filtering is often used to prevent over-segmentation. This morphological or smoothing filtering reduces noise but also removes some details that can not be segmented anymore. Secondly, as the watershed must be applied on the gradient image, edges are different regarding the chosen implementation for computing the gradient. Finally, the watershed algorithm itself leads to some inaccuracies. We will present in this paper a simple post-processing filter to automatically refine the edges between different video object planes.

The paper is organized as follows. First, we apply a succession of pre-processing filters in Section 2. There, the global motion is compensated and the image is filtered in order to reduce the over-segmentation. The segmentation is then computed in Section 3. This segmentation is based on the watershed theory. The watershed algorithm transforms the gradient image in an over-segmented image. Then, the regions are merged together before being classify in Section 4. A post-processing step for refining the edges is presented in Section 5. Section 6 shows and discuss about some results on video sequences. Finally, conclusions are drawn in Section 7.

2. PREPROCESSING

2.1. Background motion compensation

Region based image segmentation often extracts some characteristics of the image, as color or texture, and try to delimit the boundaries between those characteristics. Video segmentation adds a supplementary dimension to the problem. In this content, the segmentation algorithms can rely on the temporal coherence and the motion analysis. Many video segmentation algorithms estimate the background and extract it from the current image by a subtraction.⁸ When the camera is moving, the camera motion must be evaluated and the local movements of the objects within the sequence must be differentiated from the global background motion. This background compensation has to be achieved in a pre-processing step before segmenting. It consists in estimating and compensating the global motion due to the camera movement. The camera motion can be modelled by a mathematical transformation. We model it by a height-parameters transformation, called "pseudo-projective".⁹ It does not correspond to any physical transformation but is an approximation of the projective model. Let (x, y) be a position in frame k and (x', y') the position of the same pixel in frame $k + 1$. The model of the transformation can be written mathematically as :

$$\begin{pmatrix} x' \\ y' \end{pmatrix} = \begin{pmatrix} ax + by + c + gxy + hx^2 \\ dx + ey + f + hxy + gy^2 \end{pmatrix}. \quad (1)$$

The local motion vectors $(x' - x, y' - y)$ are estimated by using the classical full search block matching algorithm on 16x16 blocks. Figure 1 shows the result of the motion estimation in the "Coastguard" sequence and in the "Mother and daughter" sequences. Figure 1(d) shows that some errors in estimating motion vectors remain with this technique. Outlier vectors are then removed according to the filter criteria presented in¹⁰ (Figure 1(e)). The height parameters of the global motion model are then estimated by using the least square method. Small local motions can highly inference in the result of global motion, especially if the camera is not moving. To prevent detecting an erroneous global motion because of small local motions, the height parameters are compared to a

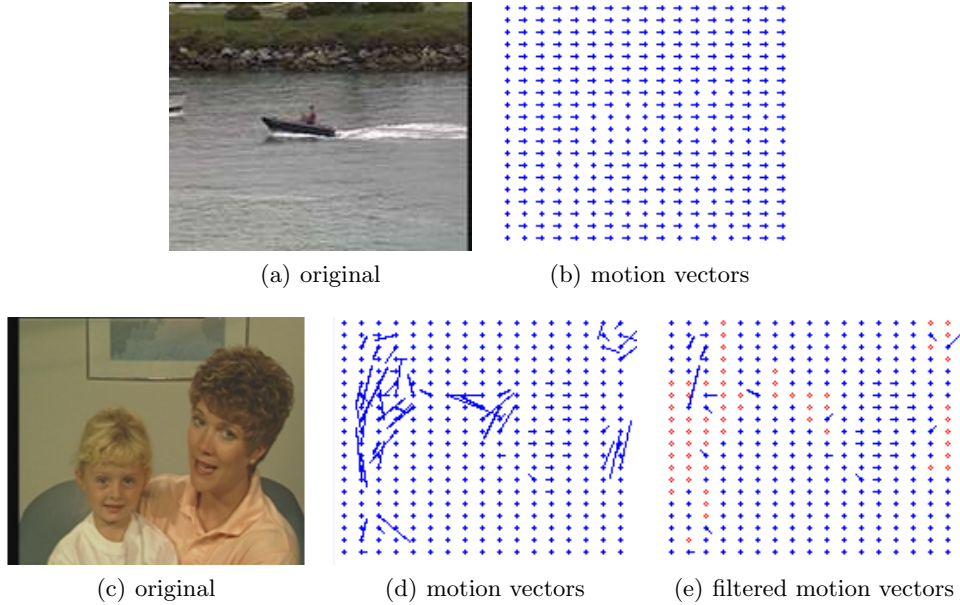


Figure 1. Motion vectors estimation

threshold. If the global motion is too small, we assume that there is no motion. Finally, the global motion is compensated according to the pseudo-projective model.

2.2. Image simplification

The main drawback of the watershed transformation is the resulting over-segmentation of the image. A first solution to diminish this over-segmentation is to filter the image before applying the watershed. The filter has for constrain to respect the edges of the objects. Mathematical morphology brings such filters. The two basic morphological operations, erosion and dilation, lead to the definition of the opening and the closing. The opening and the closing by reconstruction have the particularity to preserve the original contour.¹¹ Our morphological filter is then the succession of an opening by reconstruction and a closing by reconstruction using a flat structuring element of size 5x5.

3. SPATIO-TEMPORAL SEGMENTATION

3.1. Gradient approximation

The computation of the gradient image is a big challenge for the watershed segmentation as it remains an ill-posed problem. The goal is to detect meaningful discontinuities in the image structure. A noisy gradient will introduce a high over-segmentation. The same problem will occur with a gradient containing too many edges. On the contrary, a gradient containing few edges will not separate two different regions which have close colors. A trade off has to be done. We decide to compute the gradient in the YC_rC_b space. This way, basic over-segmented regions will be homogeneous in a color point of view and not only on a gray level point of view. We compute the gradient by using a classical formulation:

$$G(x, y) = \sqrt{\left(\frac{\partial I(x, y)}{\partial x}\right)^2 + \left(\frac{\partial I(x, y)}{\partial y}\right)^2}, \quad (2)$$

where $I(x, y)$ is respectively Y , C_r and C_b . G_Y corresponds to the gray level magnitude and G_{C_r} and G_{C_b} contains the color information. Other gradients are often used and can also perform a little filtering, like the Canny gradient or the morphological gradient.

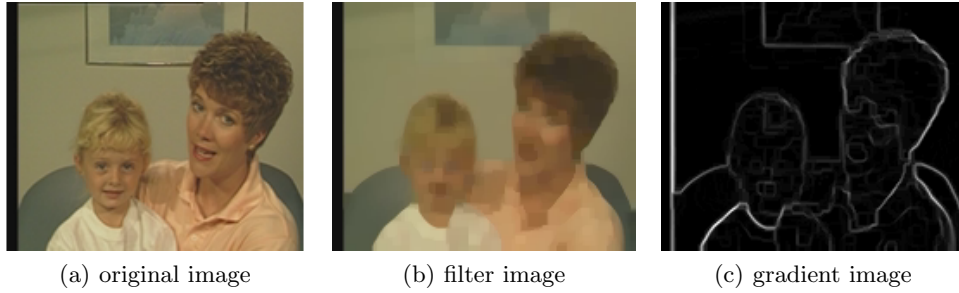
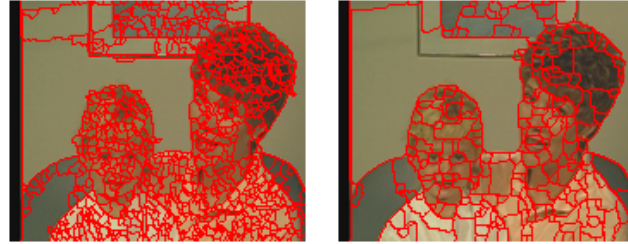


Figure 2. Filtering and gradient



(a) regions in original image (b) regions in filtered image

Figure 3. Watershed regions

Three gradients are first computed on Y , C_r and C_b . Noise is much more present in the color chrominance than in the luminance. To reduce this effect, it is common to combine the three gradients by doing a maximisation rather than a simple sum. Let w_1, w_2 and w_3 be some weight coefficients, the final gradient can be express as follow :

$$G = \max\{w_1G_Y, w_2G_{C_r}, w_3G_{C_b}\}. \quad (3)$$

The weighting factors are found experimentally. The resulting image gradient is given as input in the watershed algorithm. Figure 2 shows the resulting gradient.

3.2. Watershed transform

The watershed transformation is a tool of the morphological mathematic which is widely used for segmentation purpose. The input gradient image is treated as a topographic surface, containing hills, plateaus and valleys. We use the immersion based algorithm introduced by Vincent and Soille.¹² Each local maxima of the topographical surface leads to a region boundary. The main drawback of the watershed algorithm is its high sensitivity to noise. This sensitivity results in an over-segmentation. Meyer and Beucher¹³ have shown a method to use markers in order to reduce this over-segmentation. Our approach is to perform the fusion after the watershed. Notice that the pre-processing filtering reduces already significantly the number of regions as it is shown in Figure 3.

3.3. Fusion

We have seen in the previous section that the watershed algorithm introduces an over-segmentation. A fusion step is thus necessary to solve this problem. A very intuitive way is to first merge both regions which are the most similar according a certain criterion. This is done by constructing a Binary Partition Tree (BPT).

3.3.1. Region Adjacency Graph

For a computational consideration, it is important to sort and label all the regions created by the watershed segmentation. Each region has proper characteristics such as mean color, histogram, entropy, texture, curvature,

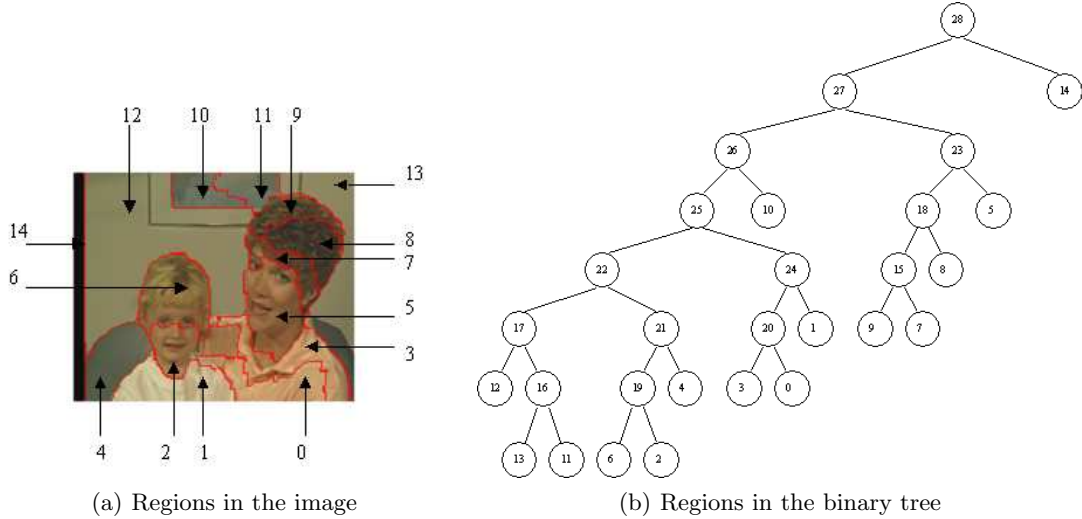


Figure 4. Binary Partition Tree

associated motion, . . . Each region is also defined by its relations with its neighbors. For instance : the neighbor color, the boundary length and the intensity of the gradient on the boundary. We can thus consider the regions, R_i , as the nodes of a Region Adjacency Graph (RAG). The set of the N regions obtained by the watershed segmentation is $R = \{R_1, \dots, R_N\}$. The connections between the regions are the edges. We say that E_{ij} is an edge if and only if R_i is a neighbor region of R_j . The set of all the edges is E . The entire graph, RAG , is composed by the set of the nodes, R , and the set of the edges, E .

3.3.2. Binary Partition Tree

To merge the over-segmented regions we need to determine a fusion order. This can be achieved by organizing the regions in a Binary Partition Trees (BPT) according to a similarity criterion. BPTs have been extensively discussed by Salembier for segmentation and compression purpose.^{14, 15} We create a binary tree in which the tree leaves are the initial partitions. When two regions are merged together, the resulting region is called the parent node and the two merged regions are called the children nodes. Children regions in the lower part of the tree will generally be very similar whereas two regions at the upper part of the tree will be rather different.

The creation of the tree goes first by the computation of the similarity criterion between each neighbor regions. Then, the two neighbor regions which are the most similar according to this criterion, are merged and this results in the creation of a new region. Step by step, the binary tree is finally created. Figure 4 shows the tree appearance on the basis of 16 initial regions.

Thanks to this graph, it is possible to fix the number of different regions in the image either by giving a priori the final number of regions, either by using markers as it will be done in section 4.2. All this merging process has already been deeply studied in previous works.^{14, 15}

3.3.3. Fusion criterion

For the creation of the BPT, we use a spatio-temporal criterion, called STC . The STC is computed for each edge E of the Region Adjacency Graph. This criterion contains a spatial part, called SC , and a temporal part, called TC :

$$STC = \alpha SC + \beta TC, \quad (4)$$

where α and β are the weighted factors. For still image segmentation, as for the first frame of a video sequence, $\beta = 0$.

The SC term measures the mean difference of the mean intensity between two adjacent regions. Let $\bar{\mathbf{I}}_{ik} = (\bar{I}^1_{ik}, \bar{I}^2_{ik}, \bar{I}^3_{ik})$ denotes the vector of the mean intensities of the three color components RGB of region i in image k :

$$\bar{I}^l_{ik} = \frac{1}{N_i} \sum_{(x,y) \in R_{ik}} I^l_k(x,y), \quad l = 1, 2, 3, \quad (5)$$

with N_i , the number of pixels within region i . The spatial criterion, SC , between two regions, R_i and R_j , is then denoted :

$$SC(ij) = \sum_{l=1}^3 \left| \bar{I}^l_{ik} - \bar{I}^l_{jk} \right|. \quad (6)$$

Let $\bar{\mathbf{D}}_{ik} = (\bar{D}^1_{ik}, \bar{D}^2_{ik}, \bar{D}^3_{ik})$ be the vector of the the mean difference between the images at time k and $k - 1$ of region i :

$$\bar{D}^l_{ik} = \frac{1}{N_i} \sum_{(x,y) \in R_{ik}} |I^l_k(x,y) - I^l_{k-1}(x,y)|, \quad l = 1, 2, 3. \quad (7)$$

The temporal criterion, TC , is then denoted :

$$TC(ij) = \sum_{l=1}^3 \bar{D}^l_{ik} - \bar{D}^l_{jk}. \quad (8)$$

This way, moving regions will not merge with non moving region (and vice versa).

The spatio-temporal criterion, STC , indicates how important is the similarity between two regions. If two regions are very similar in term of color homogeneity and temporal coherency, the STC will be low.

The main drawback of such a criterion is that it does not depend of the region size. It is therefore possible to have small regions in the upper part of the tree. To solve this problem we use the merging process described as follows.

1. Set $T_{size} = T_{sizeInit}$.
2. If there exist regions R_i , of size N_i , respecting $N_i < T_{size}$:
 - (a) Compute all the $SCT(ij)$ for all regions R_i respecting $N_i < T_{size}$.
 - (b) Find the edge E_{ij} which has the smallest $SCT(ij)$ and merge the two corresponding regions:

$$E_{ij} = \arg \min_{R_i: N_i < T_{size}} SCT(ij) \quad (9)$$

- (c) Go back to step 2.

3. $T_{size} = T_{size} + T_{step}$
4. If $T_{size} > FrameSize$, stop. Otherwise, go back to step 2.

Selection of the thresholds $T_{sizeInit}$ and T_{step} allows us to control the sensitivity of the merging process. We can remark that it is possible to not consider the region size during the binary tree creation (for instance, by using $T_{sizeInit}$ and T_{step} respectively equal to the $FrameSize$ and 1).

4. CLASSIFICATION

4.1. Temporal classification

Statistical hypothesis tests are often used in video processing to determine a change detection mask (CDM). The CDM is a binary image indicating, for each pixel, if it has changed or not in the current image with respect to the previous image. If two successive frames are completely static, without illumination change, the only difference that we can expect between successive intensities are due to noise. This noise can be modelled as a normal distribution.¹⁶

Let $D_k^l(x, y) = I_k(x, y) - I_{k-1}(x, y)$ be the image difference of intensities between image I_k and image I_{k-1} . $D_k^l(x, y)$ is the luminance difference. To this difference corresponds a zero-mean Normal distribution. We perform an equality test on two variances of the image difference. The first variance, σ_1 , is estimated by S_1 on n_1 random samples, X_1, X_2, \dots, X_{n_1} , in an area that we know being static. The other variance, σ_2 , is estimated by S_2 on n_2 random samples in the area under consideration. The resulting random variables :

$$(n_i - 1)S_i^2/\sigma_i^2 = \sum_{j=1}^{n_i} (X_j - \bar{X})/\sigma_i^2 \quad i = 1, 2 \quad (10)$$

have a chi-squared distribution with $n_i - 1$ degrees of freedom. Let's consider a square sliding window, if the pixels in the sliding window are static, the variance within the window will be close to the reference variance of the noise. On the contrary, if an area is moving, the variance will be higher than the one of noise. Therefore, the idea is to use a statistical hypothesis on the homogeneity of variances.

$$\begin{aligned} H_0 : \sigma_1^2 &= \sigma_2^2 \\ H_1 : \sigma_1^2 &< \sigma_2^2. \end{aligned} \quad (11)$$

One can show that the statistic test is :

$$F_{n_1-1; n_2-2} = \frac{S_1^2/\sigma_1^2}{S_2^2/\sigma_2^2} = \frac{S_1^2}{S_2^2}, \quad (12)$$

under the null hypothesis. The random variable $F_{n_1-1; n_2-2}$ is a Fisher distribution with $(n_1 - 1, n_2 - 1)$ degrees of freedom. The null hypothesis will be rejected if $F_{n_1-1; n_2-2} > F_{n_1-1; n_2-1; 1-\alpha}$.

Doing so, a change detection mask is computed for all the pixels in the image. Changing areas are classified as foreground while static areas are considered as background. It is important to notice one of the main drawback of the CDM. The frame difference allows to detect changing in textured area. An uniform moving area will be marked as changing only on its edges. Moreover, when a moving object is revealing the background, this region will be marked as changing. It thus exists a possibility for the background, newly revealed, to be considered as foreground. All these considerations lead us to say that a region is foreground only if 10% of the pixels within it are considered as changing by the CDM.

4.2. User guided classification

For still images, it is impossible to detect foreground objects by means of the change detection mask. Markers are thus required. The user has to specify the class of some regions by drawing two lines in the image, one for each of the foreground regions and the other for the background. Markers are then propagated in the binary partition tree by a classical propagation scheme in the way of Salembier.^{14, 15}

5. POST-PROCESSING

The gradient estimation and the watershed algorithm introduce an incertitude of one pixel on the edges of the computed mask. However, it may happen that noise in the image raises on imprecisions on the mask contours. For this reason, we use a post-processing step to refine the edges where needed. Our algorithm checks each pixels close to the edges of the mask. If the color of the processed pixel is most similar to the mean color of a neighbor region than to the mean color of its current regions, we change the pixel from region. The algorithm can be

written as follows. Let p be a pixel, Img the image,

Edge refinement algorithm

```

//Init Queue Q
for all p in Img
if  $\exists$  one  $q$ , neighbor of  $p$ , such that class  $p \neq$  class  $q$ 
then add  $p$  to  $Q$ 
end for

//loop
while  $Q$  is !Empty

1.  $p \leftarrow$  first pixel of the queue
2.  $tmp_1 \leftarrow$  false,  $tmp_2 \leftarrow$  true,  $tmp_3 \leftarrow$  true
3. for all  $q$ , neighbors of  $p$ ,
   (a) if ! $tmp_1$ 
       i.  $tmp_2 \leftarrow CDM_q$  // CDM is 0 (background) or 1 (foreground) background
       ii.  $tmp_1 \leftarrow$  true
   (b)  $tmp_3 \leftarrow$  !( $tmp_2$  xor  $CDM_p$ ) and  $tmp_3$  // if  $tmp_3$  is true : all the neighbors pels have the same CDM
   // if the pel is surrounded by only one single layer : change the pel layer if it belongs to the other region,
   and remove it from the set.
4. if  $tmp_3$  then change pel region if region  $p \neq$  region of neighbors
5. else, for all  $q$ , neighbors of  $p$ ,
   (a) Compute  $Sim(p, regp)$  // compare the mean color of region  $p$  with the color of  $p$ 
   (b) Compute  $Sim(p, regq)$  // compare the mean color of region  $q$ , neighbor of  $p$ , with the color of  $p$ 
   (c) if  $Sim(p, regq) > Sim(p, regp)$  then change pel region

end while

```

For the similarity criterion, we use the L_2 norm on the RGB colors :

$$Sim(x, y, i) = \sqrt{\sum_{l=1}^3 (I_k^l(x, y) - \bar{I}_{ik}^l)^2}, \quad (13)$$

where $I_k^1(x, y), I_k^2(x, y), I_k^3(x, y)$ are the red, green and blue color intensity of pixel (x, y) in image k and $\bar{I}_{ik}^1, \bar{I}_{ik}^2, \bar{I}_{ik}^3$ are the mean red, green and blue color intensity of region i in image k .

As the edge refinement do not always converge through the wanted mask, the choice of the application of the process is let to the user. Figure 6 shows the improvement of such a process.

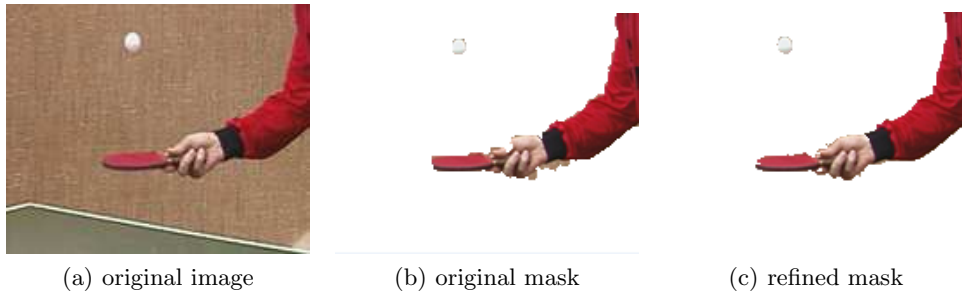


Figure 5. Edge refinement



Figure 6. Mask results

6. RESULTS

The results are quite promising with respect to the complexity of the whole process. As we can see in Figure 6(a), the algorithm is well suited for segmenting big objects laid over uniform background, not too much textured. We can also notice, in Figure 6(b) that the preprocessing filter reduces noise in the image but has one major drawback. It is then more difficult to segment small objects. Figure 7 shows the results on several frames of the sequence "Mother and daughter". The mask is obtained by a supervised segmentation, requiring less than six user interaction by frame.

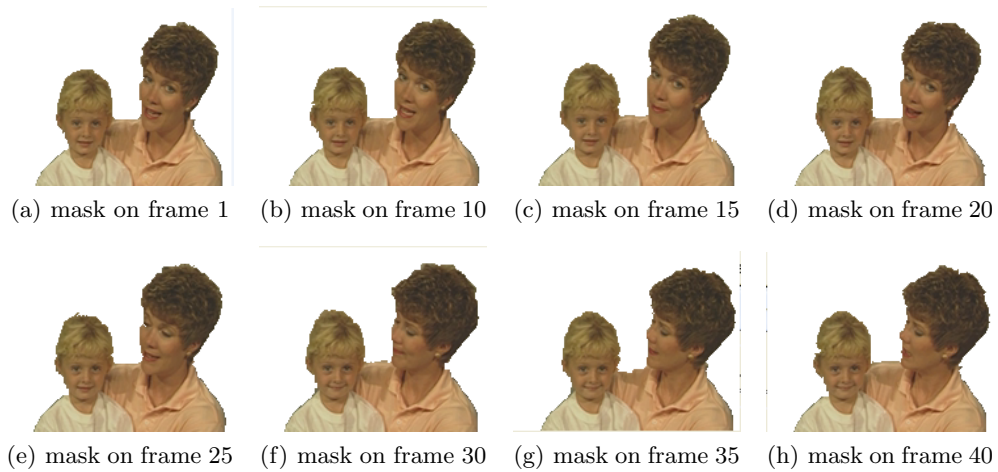


Figure 7. Mother Sequence

7. CONCLUSION AND FUTURE WORKS

We proposed an accurate supervised segmentation method for post-production processing. The images are first simplified by means of a pre-filtering step and the global motion is compensated. The spatial segmentation of the image in homogeneous color regions by the watershed algorithm results in an over-segmented partition. A fusion step is then performed using a binary partition tree graph to determine the fusion order. Finally, the classification is done by using markers specified by the user and by using the Change Detection Mask. A post-processing filter is also proposed for refining the mask contours. The results of our method show good performance for giving an accurate segmentation mask for images with smooth background. Future works will concentrate on the tracking part. This will allow us to have a better temporal continuity of the mask and to avoid user interactions on each frame of the sequence.

ACKNOWLEDGMENTS

The authors would like to acknowledge the Belgian Region Wallonne for providing the financial support for this research.

REFERENCES

1. L. Chiariglione, "Mpeg and multimedia communications," *IEEE Trans. on Circuits and Systems for Video Technology* **7**, pp. 5–18, February 1997.
2. S.-F. Chang, T. Sikora, and A. Puri, "Overview of the mpeg-7 standard," *IEEE Trans. on Circuits and Systems for Video Technology* **11**, pp. 688–695, June 2001.
3. H. Xu, A. Younis, and M. Kabuka, "Automatic moving object extraction for content-based applications," *IEEE Trans. on Circuits and Systems for Video Technology* **14**, pp. 796–812, June 2004.
4. V. Mezaris, I. Kompatsiaris, and M. Strintzis, "Video object segmentation using bayes-based temporal tracking and trajectory-based region merging," *IEEE Trans. on Circuits and Systems for Video Technology* **14**, pp. 782–795, June 2004.
5. L. Patras, E. Hendriks, and R. Legendijk, "Video segmentation by map labeling of watershed segments," *IEEE Trans. on Pattern Analysis and Machine Intelligence* **23**, pp. 326–332, March 2001.
6. M. Kim, J. G. Choi, D. Kim, H. Lee, M. H. Lee, C. Ahn, and Y. Ho, "A vop generation tool: Automatic segmentation of moving objects in image sequences based on spatio-temporal information," *IEEE Trans. on CSVT* **9**, December 1999.
7. Y. Tsaig and A. Averbuch, "Automatic segmentation of moving objects in video sequences : A region labeling approach," *IEEE Trans. on Circuits and Systems for Video Technology* **12**, July 2002.
8. A. Monnet, A. Mittal, N. Paragios, and V. Ramesh, "Background modeling and subtraction of dynamic scenes," *IEEE International Conference on Computer Vision* **2**, p. 1305, October 2003.
9. D. Delannay, J.-F. Delaigle, and B. Macq, "Integrated fingerprinting in secure digital cinema projection," *Applications of Digital Image Processing XXIV Proceeding of SPIE* **4472**, 2001. San Diego, CA.
10. A. Dante and M. Brookes, "Precise real-time outlier removal from motion vector fields for 3d reconstruction," *International Conference on Image Processing* **1**, pp. 393–396, 2003.
11. P. Salembier and M. Padras, "Hierarchical morphological segmentation for image sequence coding," *IEEE Trans. on Image Processing* **3**(5), pp. 639–651, 1994.
12. L. Vincent and P. Soille, "Watersheds in digital spaces: An efficient algorithm based on immersion simulations," *IEEE Trans. on Pattern Anal. and Machine Intell.* **13**, pp. 583–598, 1991.
13. F. Meyer and S. Beucher, "Morphological segmentation," *Journal of Visual communication and Image representation* **1**(1), pp. 21–46, 1990.
14. P. Salembier and L. Garrido, "Binary partition tree as an efficient representation for image processing, segmentation, and information retrieval," *IEEE Trans. on Image Processing* **9**(4), p. April, 2000.
15. P. Salembier and F. Marques, "Region-based representations of image and video: Segmentation tools for multimedia services," *IEEE Trans. on Circuits and Systems for Video Technology* **9**, December 1999.
16. T. Aach and A. Kaup, "Statistical model-based change detection in moving video," *Signal Processing* **31**, pp. 165–180, 1993.