Label Prediction and Local Segmentation for Accurate Video Object Tracking

Guillaume Foret, Pascal Bertolino

Label Prediction and Local Segmentation for Accurate Video Object Tracking

Guillaume Foret, Pascal Bertolino

Laboratoire des Images et des Signaux,
BP 46, 38402 Saint Martin d'Hères, France
Guillaume.Foret, Pascal.Bertolino@lis.inpg.fr

ABSTRACT

This paper presents an approach dedicated to accurately track one or several semantic objects in a video sequence. The accurate tracking of the partition object boundary is obtained by a label prediction. This prediction is performed thanks to motion vectors obtained with two different block-matching uses. In the predicted partition, a local segmentation is necessary only where matching failed and close to the predicted boundaries, in order to get the most accurate boundaries. This local segmentation is then followed by a classification step. During the classification a backward projection is used to assign or not a region to a given object.

Keywords: tracking, label prediction, local segmentation, irregular pyramid, block-matching, non-rigid objects.

1. INTRODUCTION

The development of object-based video manipulation needs more and more techniques to accurately extract and track video objects in natural video sequences. Normally the video object extraction includes two steps: the object definition and the object tracking. We suppose in this paper that the video object definition is already performed and we focus on a method to track with accuracy the object during its evolution in the video sequence.

Many methods use homogeneous gray scale/color as a criterion to track regions. Some of them are characterized by a forward projection[7, 11, 13]. Once the partition $P(t)$ of the frame $F(t)$ is available, this partition is motion compensated and spatially adjusted to obtain the next partition $P(t+1)$. In[7, 13], the video object is tracked by updating its boundary in each new frame. The weak point of these methods is their difficulties to deal with disconnected video objects or non-rigid motions. In[11, 15], the authors define a video object as a group of spatial-homogeneous regions and track all of them. The main drawback of this method lies in the fact that very small regions cannot be used in the definition of the object. This constraint may entail some difficulties to accurately define the object on its boundary.

To preserve the object boundary accuracy during the tracking, we have to consider in each new frame all the physical edges (spatial discontinuities). Indeed the video object boundary is located on these edges. Several methods suggest to first apply an independent spatial segmentation on each new frame. Then they classify each segmented homogeneous region according to the previous video object partition[1, 5, 6, 10, 14]. All these methods require an over-segmentation to keep any physical edges of meaningful entities.

Our approach combines forward projection, local segmentation and classification. It takes the advantage of an over-segmentation close to the predicted object boundary, to reduce the calculation time and to increase the reliability of the boundary. We present in the next section a brief description of our method. We develop the key points of the method in sections 3 and 4. Some results are then provided in section 5 before conclusions.

The first author wants to thank Prof. M.T. Sun (Information Processing Laboratory, Department EE, University of Washington, Seattle) and Dr. C. Gu (Microsoft Corporation, Redmond) for various interesting discussions and fruitful cooperation during his stay at the Information Processing Laboratory.

2. OVERVIEW OF THE METHOD

2.1. Motion estimation and label prediction

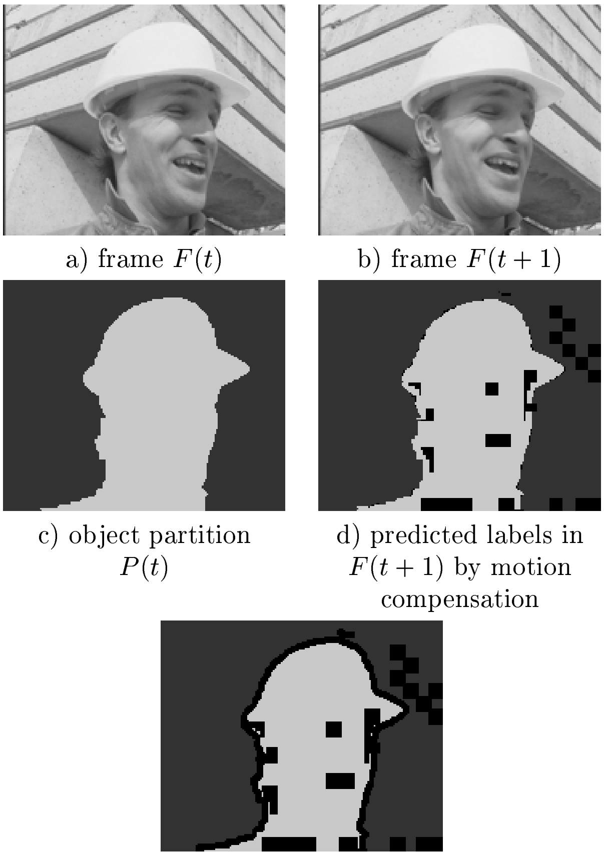
Let's consider two successive frames (fig 1.a and .b) and the already known object partition $P(t)$ of the first one (fig 1.c). In our example, the partition consists of two labels that define the object pixels and the rest of the scene (background).

A motion estimation is performed between the original frames $F(t)$ and $F(t+1)$. Like in[10], it is based on the block-matching algorithm. In order to increase the number of matches and their quality, we combined two block-matching processes (see section 3).

The motion vectors obtained are applied to the known partition $P(t)$ to predict as much as possible a label for each pixel in $F(t+1)$, that is to build a first approximation of $P(t+1)$. Figure 1.d shows the predicted labels in $F(t+1)$. Black pixels stand for non predicted pixels (unlabeled areas).

By predicting labels in $F(t+1)$ we obtain an approximation of the object boundary. Pixels close to this approximate boundary must be re-segmented to ensure the most accurate object contour. These pixels are marked unlabeled. Besides, to be less sensitive to the eventual motion estimation errors, unlabeled areas are dilated (fig 1.e).

Thanks to this label prediction a large number of pixels are already (correctly) classified. We will present in the next paragraph how the unlabeled areas are segmented to finally extract the entire video object in $F(t+1)$.



a) frame $F(t)$

b) frame $F(t+1)$

c) object partition $P(t)$

d) predicted labels in $F(t+1)$ by motion compensation

e) final predicted labels in $F(t+1)$

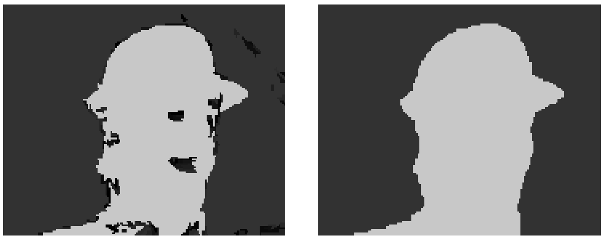
Figure 1. Label prediction (black pixels are unlabeled pixels)

2.2. Local segmentation

This step is performed after the label prediction. Its goal is to over-segment in homogeneous regions the remaining unlabeled areas in $F(t+1)$ (pixels without predicted label). During the local segmentation process, the neighborhood of the unlabeled pixels is involved as well, so that unlabeled pixels may easily re-stick to the existing partition. In this way, the segmentation step allows many tiny regions to be labeled (i.e. directly classified either as object or as background). Figure 2.a shows the result of the over-segmentation on the unlabeled areas defined in figure 1.e. The spatial segmentation method used is more detailed in Section 4.

2.3. Classification

Even after segmentation, some regions still remain unlabeled. The goal of the classification is to give each of them either the label of an object or the label of the background: each region still unlabeled is projected on the previous object partition $P(t)$ to be assigned to the object or to the background[5, 6, 14]. Like in[6], we use a simple translational motion model to estimate the motion vector of a region. We classify the region according to the label recovered in majority in $P(t)$ after the projection. This classification process provides a final object partition $P(t+1)$ for the frame $F(t+1)$. A simple morphological post-processing to smooth the boundary and to improve the visual quality is then applied to the final partition (fig 2.b).



a) local segmentation result

b) final object partition $P(t+1)$

Figure 2. Next object partition building (in (a) unlabeled regions are represented by different dark gray levels)

3. ABOUT THE MOTION ESTIMATION METHOD

Our label prediction is based on the well known block-matching algorithm used in many video coding applications.

3.1. Block-matching algorithm

In order to estimate motion between two frames F_1 and F_2, this algorithm manipulates square blocks of several pixels (8×8 pixels for QCIF format videos). The motion vector of a specific block in the frame F_1 is obtained by searching the best matching block within a search area in F_2. The matching criterion used is the Sum of Absolute Differences (SAD). The SAD of two $N \times N$ blocks X and Y ($X \in F1$, $Y \in F2$) is defined as:

$$SAD(X,Y) = \sum_{i=1}^{N} \sum_{j=1}^{N} |X(i,j) - Y(i,j)| \qquad (1)$$

For a given source block, the best matching block is the block which minimizes the SAD within the search area. Two blocks match or not according to the gray level repartition of their pixels. If they match, it is assumed that they correspond to the same entity in the two frames to infer a motion vector. We have to notice that motion vectors obtained by matching blocks that include parts of different regions (i.e. that include discontinuities) are more reliable than those obtained by matching blocks mainly composed by homogeneous gray level values.

A particularity that makes both the strength and the weakness of the block-matching technique is the assumption that all pixels within a block undergo a uniform motion, namely a translation. This hypothesis may be restrictive when rotations occur for instance, or for blocks that include several regions which might have different motions. But most of the time, since motion in videos is low, heterogeneous blocks manage to match correctly. Besides that, blocks which don't satisfy at all this hypothesis can be rejected because they will only match with a high SAD.

Many algorithms performing block-matching can be found in the literature (a recent and synthetic review can be found in[3]). We use the Block Sum Pyramid Algorithm (BSPA)[8]. It is based on a fast motion estimation method called the Successive Elimination Algorithm (SEA)[9], which achieves the same estimation accuracy as the Full Search Algorithm (FSA) while requiring less computation time.

3.2. Prediction by regular block-matching

Let us consider two successive frames $F(t)$ and $F(t+1)$. We can build a prediction of $F(t+1)$ from $F(t)$ with the regular backward block-matching between $F(t+1)$ and $F(t)$: according to a regular grid, the algorithm divides $F(t+1)$ into square source blocks of pixels. For each source block, a search is conducted within a confined window in $F(t)$ to locate the best matching block. Then a prediction of $F(t+1)$ is approached by replacing each source block by the corresponding one from $F(t)$.

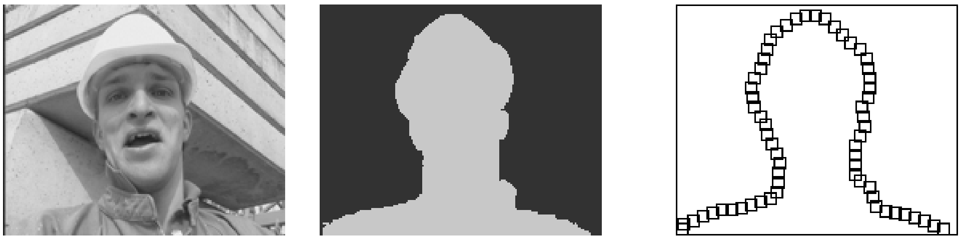
Of course, some differences (called prediction errors) may occur between the real $F(t+1)$ and its prediction. Backward is preferred to forward block-matching since it produces more simple predictions: every source pixel gets one and only one prediction.

3.3. Local block-matching principle

During the regular block-matching, the division into source blocks of $F(t+1)$ is classically applied on a regular grid. In all our experiments, using a subset of source blocks centered on the edges of the objects increased the quality of the edge prediction. We call this new block-matching the *local block-matching*.

This matching is still performed using two successive original images $F(t)$ and $F(t+1)$. Only the choice of the blocks to process and their localization is different. In order to locate the blocks over the edges of the objects, both the original source image and a good localization of its edges (in the form of a partition for instance) are required.

Homogeneous blocks are not processed. To accurately locate edge blocks, the borders of the partition are traversed and blocks are regularly disposed along them[4]. To increase the possibility of each part of the boundary to be matched, the source blocks may slightly be superimposed (figure 3).



a) original frame b) partition of the frame c) location of the source blocks

Figure 3. Example of how blocks are located for the local block-matching

3.4. Local vs. global block-matching

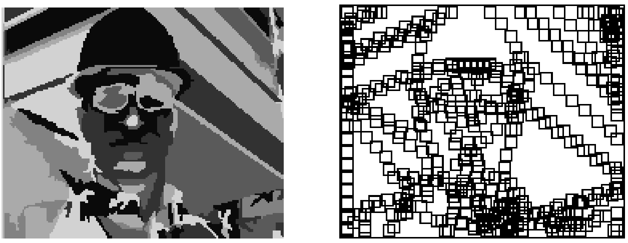
An objective comparison is necessary to show that local block-matching provides a better edge prediction than regular block-matching. With two successive frames, measuring the quality of the two predictions close to the region edges is a good way to make this comparison. Moreover, it can be performed over the whole sequence to give an idea of the quality variation according to the sequence content.

In order to compare the two methods in the same experimental conditions, only block-matching in backward mode is used. This implies that each $F(t+1)$ edge set must be known *a priori* so that the blocks may be located correctly in the case of the local block-matching. For this reason, each frame $F(t+1)$ is independently spatially segmented in homogeneous regions to provide the edge set location.

A first prediction $P1$ using the regular block-matching is performed. Only blocks containing edges in $F(t+1)$ will be taken into account for the comparison.

A second prediction $P2$ is achieved using the local block-matching (figure 4).

The quality comparison is performed by calculating the mean square error close to edges between the original frame $F(t+1)$ and each of the two predictions $P1$ and $P2$. In $F(t+1)$, $P1$ and $P2$, pixels that are involved in the computation of the error are all the couples of pixels that are located across the edge set.



a) partition in homogeneous regions (in false color)

b) source blocks layout

Figure 4. Example of local blocks location used for the comparison of the 2 block-matching methods

Figure 5 shows the results of different comparisons by using independently each block-matching. On all the sequences processed, the error is lower or equal with the local block-matching. Besides, the more important the motion in the sequence, the higher the variation between the two errors is.

3.5. Label prediction

In order to predict partitions, the motion vectors obtained during the original image block-matching are then applied to the corresponding blocks of the partition itself to predict the next partition.

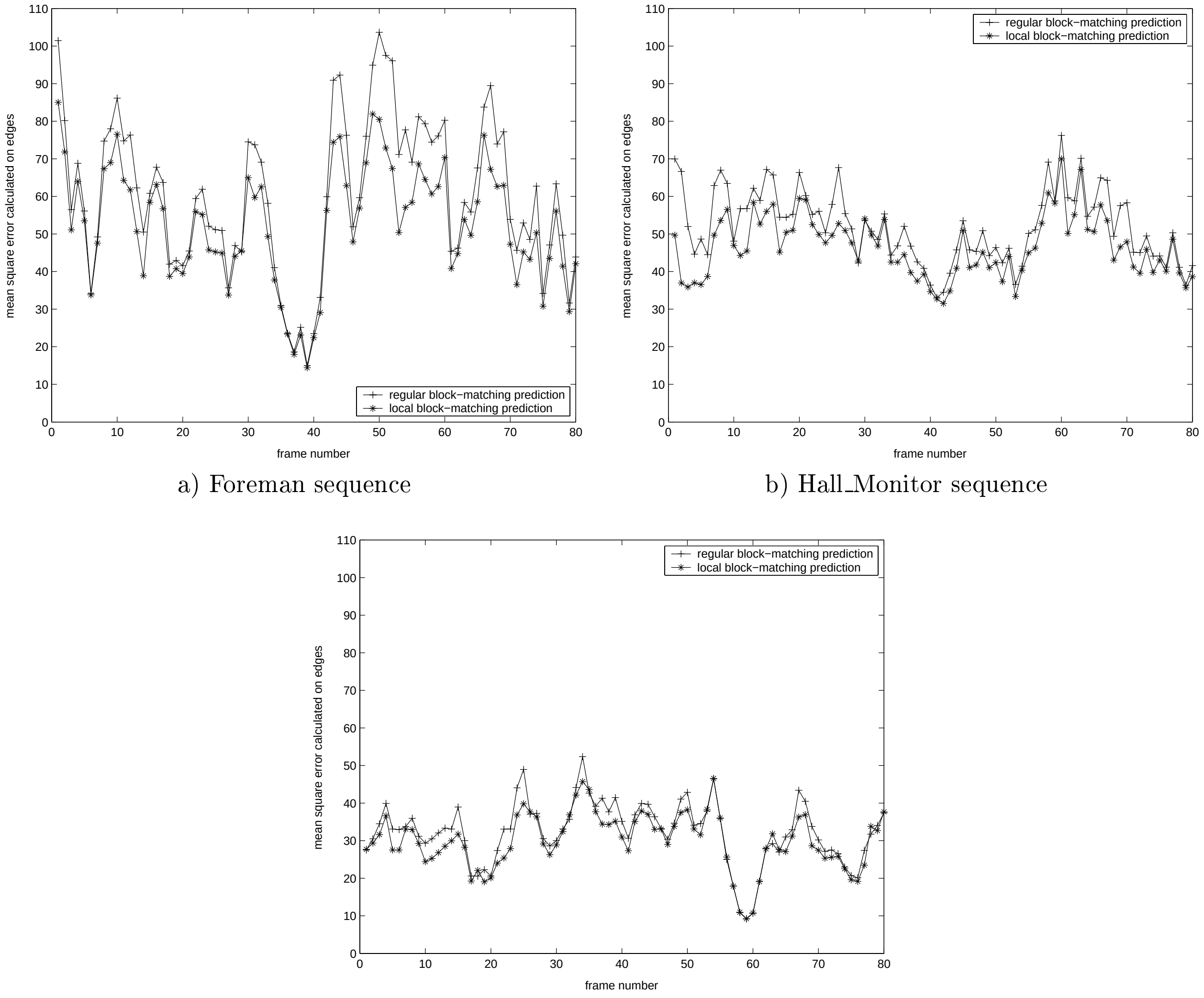
In other words, $P(t+1)$ is predicted with blocks of $P(t)$ using motion vectors computed between $F(t)$ and $F(t+1)$. Finally, the partition is predicted block by block. Each block is either homogeneous (all its pixels have the same label) or not (the block is located on the border of two or more objects).

In our method we decided to combine the regular and local block-matching to predict labels. The regular block-matching is used in backward mode to predict as much as possible a label for each pixel in $F(t+1)$. Then a local block-matching provides a more accurate predicted label for the pixels close to the video object boundary. Since the local block-matching is performed according to the video object boundary defined in the partition $P(t)$, that means, we have to use this algorithm in forward mode.

During the motion estimation in order to avoid attributing a wrong motion vector to a block, a similarity threshold is used: the motion vector obtained with two blocks that provide the minimum SAD may indeed be used for the prediction if the value of the SAD is less than this threshold. Otherwise, the motion vector cannot be used. This entails some non predicted zones. We fix the same similarity threshold for the two kinds of block-matching.

Figure 6 highlights the label prediction robustness in case of a strong motion. To illustrate it, we voluntarily achieved the process with two frames that are not successive, namely $F(t)$ and $F(t+3)$ (fig 6.a and .c). We apply our method to predict labels in the second frame according to the available partition of the first one. We provide as an intermediate result the prediction obtained by using only the regular block-matching (fig 6.d). Figure 6.e shows the improvement obtained by using in addition the local block-matching. Indeed this method increases both the number of matches close to the object boundary and their quality.

This prediction is then processed to be less sensitive to estimation errors (fig 6.f) and used to create the desired object partition (fig 6.g).



a) Foreman sequence

b) Hall_Monitor sequence

c) Mother&Daughter sequence

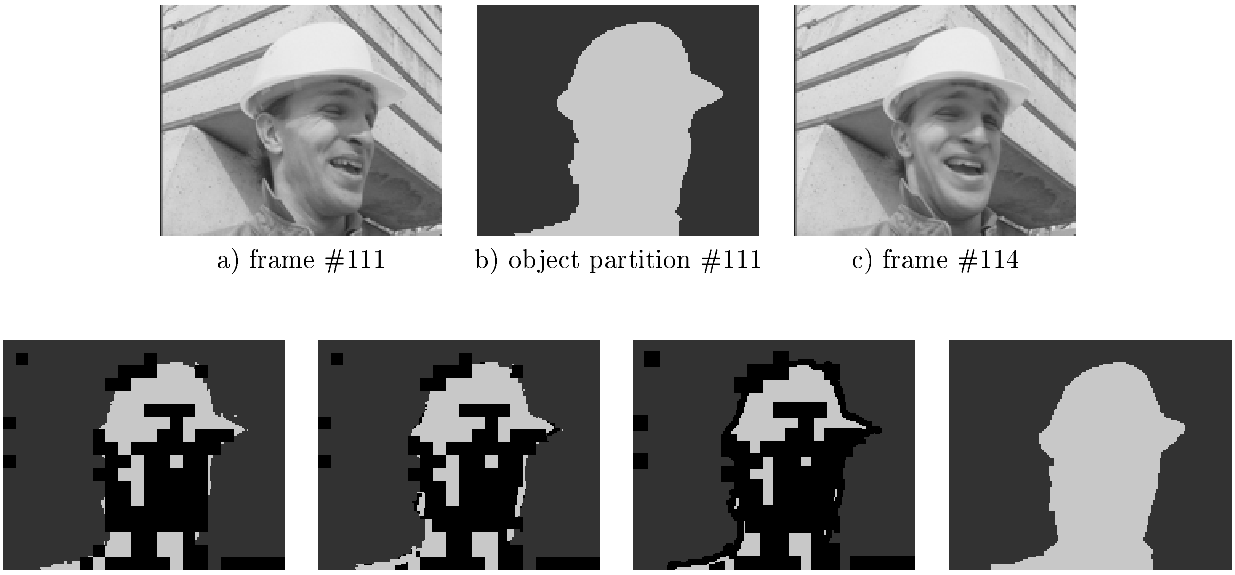
Figure 5. Mean square error on edges over 80 frames for three different sequences

4. ABOUT THE SEGMENTATION METHOD

The irregular pyramid[12] can be used to spatially segment a whole frame or just some parts of it. This is a particular region growing segmentation technique. Contrary to the watershed algorithm[5, 10], it does not need any initial markers. Its data structure is dedicated to its particularities: each level of the pyramid is represented both with an adjacency graph and a partition.

Once built, the pyramid is a stack of partitions of the original image[2], of decreasing resolution. Each new level is obtained by merging in parallel similar adjacent regions. The pyramid construction stops when no more adjacent regions can merge according to the similarity criterion. Its graph representation is well suited both to re-stick a set of pixels to a main partition and to segment any arbitrary-shape areas.

In our case the similarity criterion is related to the luminance and chrominance components. The use of the color information ensures that the final regions present color homogeneity. Two adjacent regions are considered similar if their distance in color is lower than a fixed threshold (*color_threshold*). The distance is defined as:


a) frame #111 b) object partition #111 c) frame #114

d) predicted labels by regular block-matching e) predicted labels by regular and local block-matching f) final predicted labels in frame #114 g) object partition #114 after segmentation

Figure 6. Label prediction robustness

$$d(R_1, R_2) = \sqrt{(y_1 - y_2)^2 + \gamma.[(u_1 - u_2)^2 + (v_1 - v_2)^2]} \tag{2}$$

where (y_i, u_i, v_i) represents the YUV color components of the region R_i. γ is a normalization coefficient used to compensate the difference of scale between the luminance component and the two chrominances. It is automatically calculated on the first frame according to the width of its histograms:

$$\gamma = min(\frac{y_{max} - y_{min}}{u_{max} - u_{min}}, \frac{y_{max} - y_{min}}{v_{max} - v_{min}}) \tag{3}$$

Over-segmentation is necessary to keep very homogeneous regions and to avoid loss of meaningful edges.

In our approach the spatial segmentation occurs in two cases:

(i) to fill areas of the image that could not be predicted: many reasons may explain what happened between the two frames, mainly: a motion too strong for the block-matching algorithm, a local strong change or deformation, the appearance of an object. According to the similarity criterion, the corresponding region may stick to a neighboring object (in other words the region is absorbed by the object). Obviously, the pixels of this region inherit the object label.

If the similarity criterion does not allow the fusion of the region with any object, the region is considered as unlabeled.

(ii) to fill the narrow strip of pixels located over the object boundaries. Again, the segmentation can decide to stick a pixel to a neighboring object or to create an unlabeled region.

Unlabeled regions are processed in the classification step (section 2.3).

5. RESULTS

In this section we present some results obtained on four MPEG sequences in the QCIF format: Carphone, Foreman, Coastguard and Mother&Daughter. For these sequences the initial object partitions (fig 7.e, fig 8.e, fig 9.e and fig 10.a) are manually obtained with an interactive user interface. The main segmentation parameters are: $color\_threshold = 7$ levels of the components and the minimum size for a region: 5 pixels. The same set of parameters is used for all the results.

Figure 7 and 8 show the accurate tracking of a non-rigid object. Figure 9) presents the tracking of an object composed by a large number of small homogeneous regions. We can observe in these results the robustness of our method to track edges that are poorly contrasted on the video object boundary (cf edges between the boat extremities and the water).

Last result (figure 10) gives the tracking result of several objects initially defined in the figure 10.a.

Those results were obtained on a 1 GHz Pentium PC. Each frame is processed in less than one second.

6. CONCLUSION

The label prediction step provides several advantages: firstly, a large percentage of the image surface can be tracked as is, from frame to frame, with a high confidence rate. Secondly, in the case of object appearance, lack of good matches clearly indicates which areas could be concerned. Thirdly, it reinforces the tracking of the less contrasted parts of the object boundary.

The local segmentation reduces the calculation time. Indeed the spatial segmentation is applied on the next frame only where it is necessary. It obliges edges to be accurately updated / localized within a restricted area. It also limits the number of regions which need the classification by the backward projection.

The method can also be used to track several objects. The graph structure induced by the irregular pyramid is a powerful way to represent the interaction of the objects that evolve in the scene.

At the moment, we have limited our work on video sequences in which discovered areas can be segmented correctly using only the spatial homogeneity information. In the future, we will extend this method to take into account new appearing objects, and to increase the robustness to deal with discovered areas which can contain new elements of the object. This treatment will be done by using information from next frames.

REFERENCES

1. A.A. Alatan, L. Onural, M. Wollborn, R. Mech, E. Tuncel, and T. Sikora. Image sequence analysis for emerging interective multimedia services - the european cost 211 framework. *IEEE Transactions on Circuits and Systems for Video Technology*, 8(8), November 1998.
2. P. Bertolino and A. Montanvert. Multiresolution segmentation using the irregular pyramid. In *IEEE International Conference on Image Processing, ICIP'96*, pages 257–260, Lausanne, Switzerland, 1996.
3. Y-S. Chen, Y-P. Hung, and C-S. Fuh. Fast block matching algorithm based on the winner-update strategy. *IEEE Transactions on Image Processing*, 10(8):1212–1222, August 2001.
4. G. Foret, P. Bertolino, and D. Cibaud. Partition projection in videos by global and local block-matching. In *IEEE International Conference on Image Processing, ICIP'02*, Rochester, USA, 2002.
5. D. Gatica-Perez, M.T. Sun, and C. Gu. Semantic video object extraction based on backward tracking of multivalued watershed. In *IEEE International Conference on Image Processing, ICIP'99*, Kobe, Japan, 1999.
6. C. Gu and M.C. Lee. Semantic video object tracking using region-based classification. In *IEEE International Conference on Image Processing, ICIP'98*, Chicago, USA, 1998.
7. C. Gu and M.C. Lee. Semiautomatic segmentation and tracking of semantic video objects. *IEEE Transactions on Circuits and Systems for Video Technology*, 8(5):572–584, September 1998.
8. C.H. Lee and L.H. Chen. A fast motion algorithm based on the block sum pyramid. *IEEE Transactions on Image Processing*, 6(11), November 1997.

9. C.W. Lin, Y.J. Chang, and Y.C. Chen. Hierarchical motion estimation algorithm based on pyramidal successive elimination. In *Published on International Computer Symposium*, 1998.

10. F. Marqués and J. Llach. Tracking of generic objects for video object generation. In *IEEE International Conference on Image Processing, ICIP'98*, Chicago, USA, 1998.

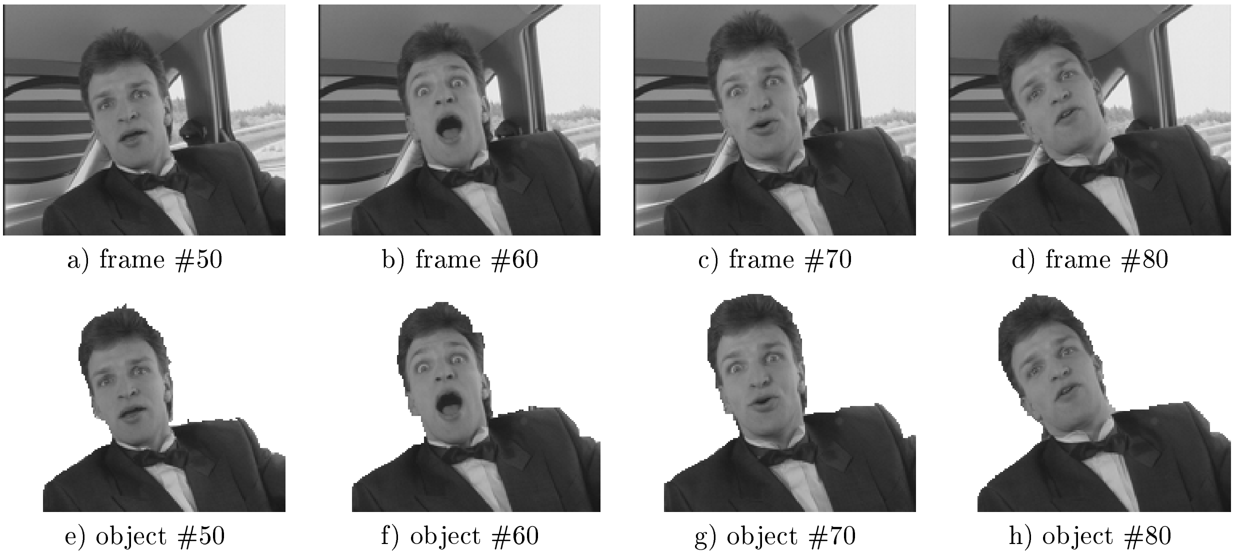
11. F. Marqués and C. Molina. Object tracking for content-based functionalities. In *SPIE Visual Communication and Image Processing, VCIP'97*, volume 3024, pages 190–199, San Jose, USA, 1997.

12. A. Montanvert, P. Meer, and A. Rosenfeld. Hierarchical image analysis using irregular tessellations. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 13(4):307–316, April 1991.

13. D.K. Park, H.S. Yoon, and C.S. Won. Fast object tracking in digital video. *IEEE Transactions on Consumer Electronics*, 46(3):785–790, August 2000.

14. S. Pateux. Tracking of video objects using a backward projection technique. In *SPIE Visual Communication and Image Processing, VCIP'00*, Perth, Australia, 2000.

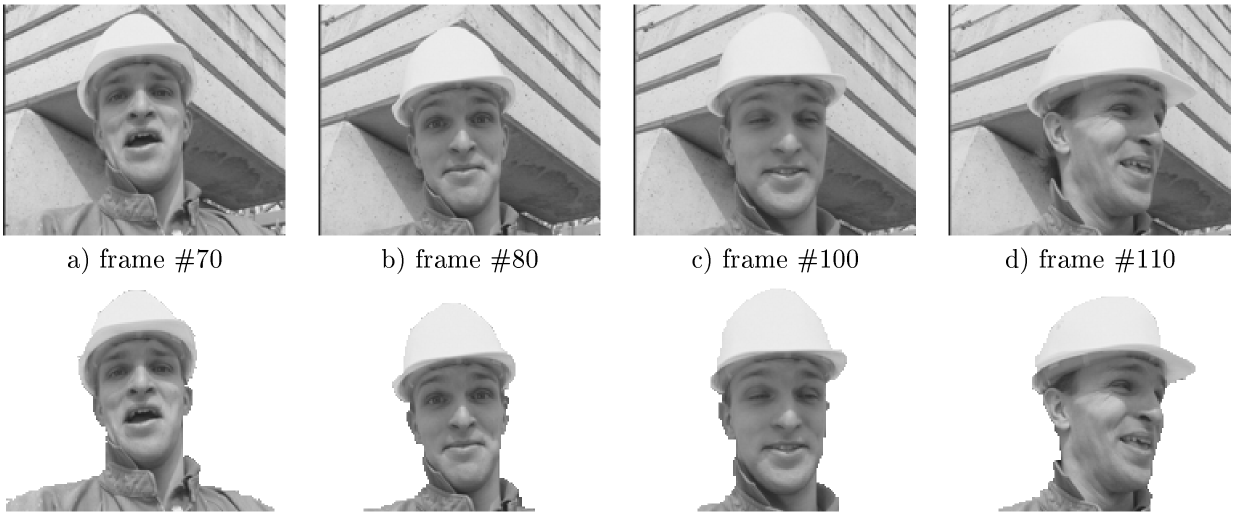
15. P. Salembier, F. Marqueés, M. Pardas, R. Morros, I. Corset, S. Jeannin, L. Bouchard, F. Meyer, and B. Marcotegui. Segmentation-based video coding system allowing the manipulation of objects. *IEEE Transactions on Circuits and Systems for Video Technology*, 7(1):60–73, February 1997.



a) frame #50 b) frame #60 c) frame #70 d) frame #80

e) object #50 f) object #60 g) object #70 h) object #80

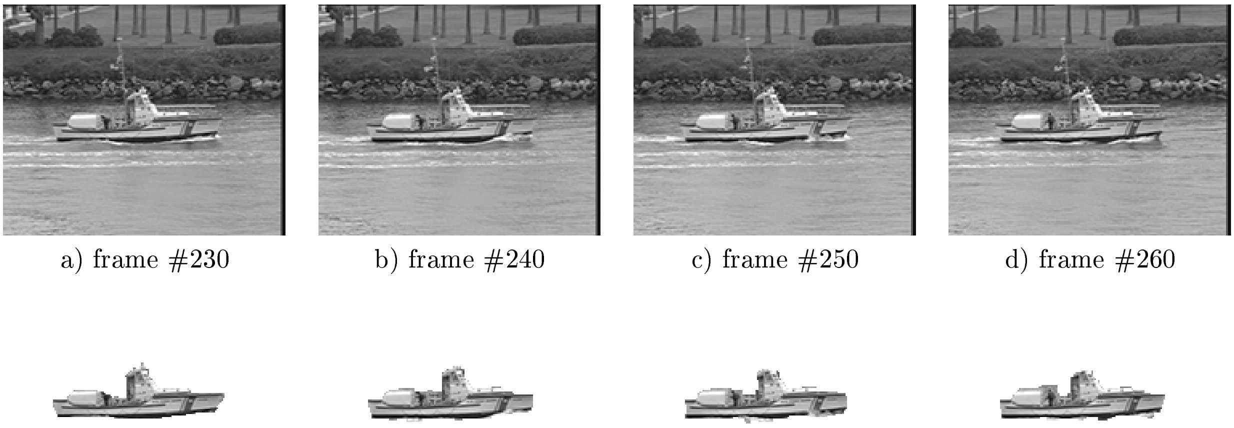
Figure 7. Non-rigid object tracking in Carphone sequence (First row presents the original frames and second row shows the video object extracted)



a) frame #70 b) frame #80 c) frame #100 d) frame #110

e) object #70 f) object #80 g) object #100 h) object #110

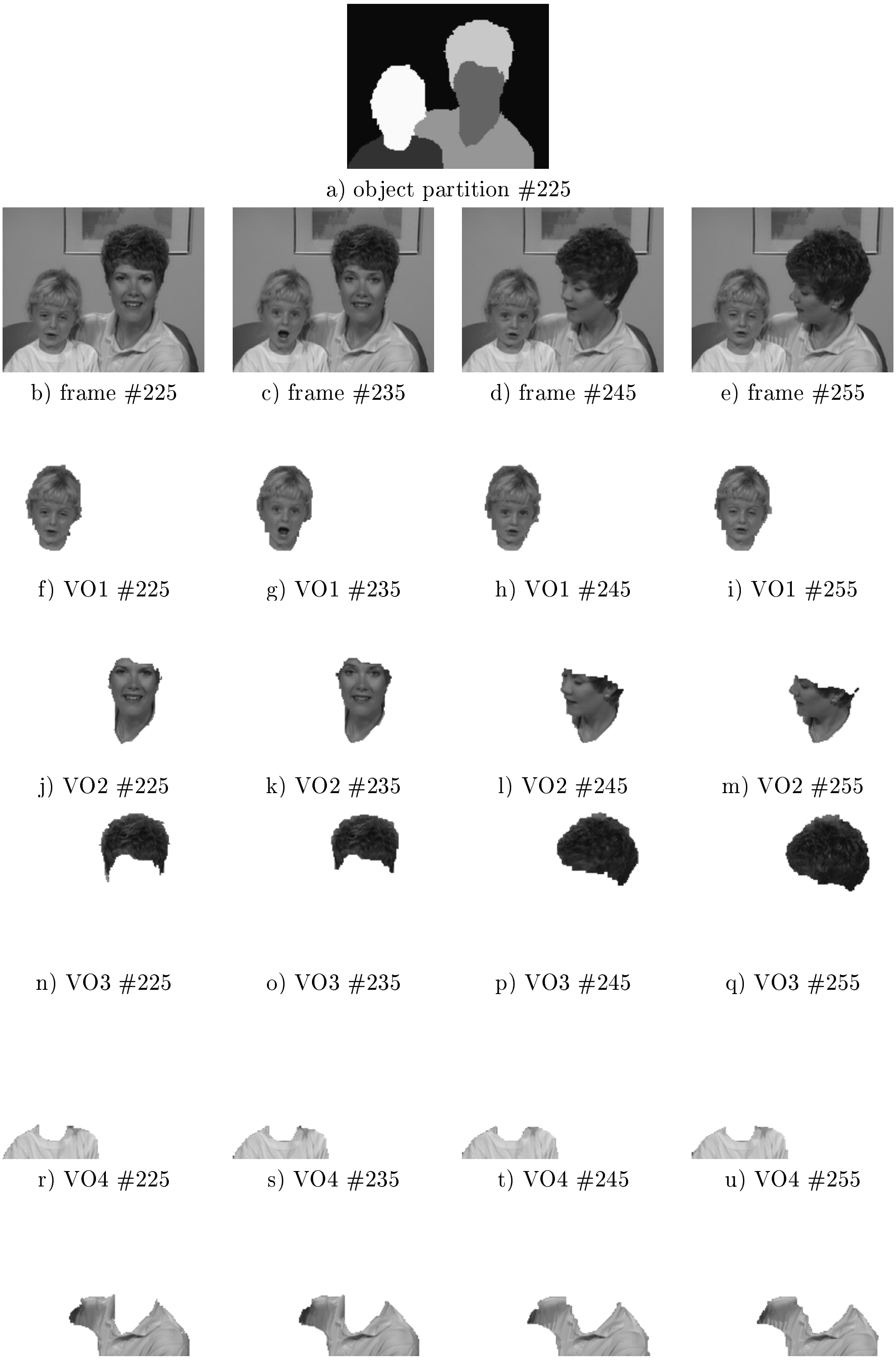
Figure 8. Non-rigid object tracking in the Foreman sequence (First row presents the original frames and second row shows the video object extracted)



a) frame #230 b) frame #240 c) frame #250 d) frame #260

e) object #230 f) object #240 g) object #250 h) object #260

Figure 9. Heterogeneous object tracking in Coastguard sequence (First row presents the original frames and second row shows the video object extracted)



a) object partition #225

b) frame #225 c) frame #235 d) frame #245 e) frame #255

f) VO1 #225 g) VO1 #235 h) VO1 #245 i) VO1 #255

j) VO2 #225 k) VO2 #235 l) VO2 #245 m) VO2 #255

n) VO3 #225 o) VO3 #235 p) VO3 #245 q) VO3 #255

r) VO4 #225 s) VO4 #235 t) VO4 #245 u) VO4 #255

v) VO5 #225 w) VO5 #235 x) VO5 #245 y) VO5 #255

Figure 10. Several objects tracked in Mother&Daughter sequence (Second row presents the original frames, next rows show the video objects extracted)