Semi-automatic Motion Segmentation with Motion Layer Mosaics

Matthieu Fradet[1,2], Patrick Pérez[2], and Philippe Robert[1]

[1] Thomson Corporate Research, Rennes, France
[2] INRIA, Rennes-Bretagne Atlantique, France

Abstract. A new method for motion segmentation based on reference motion layer mosaics is presented. We assume that the scene is composed of a set of layers whose motion is well described by parametric models. This usual assumption is compatible with the notion of motion layer mosaic, which allows a compact representation of the sequence with a small number of mosaics only. We segment the sequence using a reduced number of distant image-to-mosaic comparisons instead of a larger number of close image-to-image comparisons. Apart from computational advantage, another interest lies in the fact that motions estimated between distant images are more likely to be different from one region to another than when estimated between consecutive images. This helps the segmentation process. The segmentation is obtained by graph cut minimization of a cost function which includes an original image-to-mosaic data term. At the end of the segmentation process, it may happen that the obtained boundaries are not precisely the expected ones. Often the user has no other possibility than modifying manually every segmentation one after another or than starting over all again the process with different parameters. We propose an original easy way for the user to manually correct the possible errors on the mosaics themselves. These corrections are then propagated to all the images of the corresponding video interval thanks to a second segmentation pass. Experimental results demonstrate the potential of our approach.

1 Introduction

The problem of segmenting a video in regions of similar motion has long been an active topic in computer vision and it is still an open problem today. Motion layer extraction has many applications, such as video compression, mosaic generation, video object removal, *etc.* Moreover the extracted layers can be used for advanced video editing tasks including matting and compositing.

The usual assumption is that a scene can be approximated by a set of layers whose motion in the image plane is well described by a parametric model. Motion segmentation consists in estimating the motion parameters of each layer and in extracting the layer supports. There are many different approaches. Only examples from different classes are mentioned below.

One type of motion segmentation methods relies on the partitioning of a dense motion field previously estimated. A K-mean clustering algorithm on the motion

D. Forsyth, P. Torr, and A. Zisserman (Eds.): ECCV 2008, Part III, LNCS 5304, pp. 210–223, 2008.

estimates is proposed in [1]. The Minimum Description Length (MDL) encoding principle is used in [2] to decide automatically the adequate number of models.

Some authors [3,4] propose to combine dense motion estimation and parametric segmentation, while some others extract layers either jointly [5,6], or one after another using a direct parametric segmentation with no optical flow computation [7].

More recently, the video segmentation problem was formulated into the graph cut framework. The sequential approach described in [8] provides a segmentation map for the current image taking into account the previous labeling and matching the current image with the next one only ($t \mapsto t+1$). In such a sequential method, in poorly textured regions or when the different motions are small, motions estimated between consecutive images are unlikely to be very different from one region to another, which does not help the segmentation process.

Simultaneous batch approaches provide jointly the segmentation of N images using a 3D graph while increasing temporal consistency. Temporal constraints between an image and the subsequent ones ($1 \mapsto 2$, $1 \mapsto 3$, ...) are used in [9]. The method presented in [10] considers temporal constraints between successive images ($1 \mapsto 2 \mapsto 3 \mapsto \ \ldots$) but uses also more distant image pairs ($t \mapsto \ t+1$, $t \mapsto \ t+2$, ..., $t \mapsto \ t-1$, $t \mapsto \ t-2$, ...) to handle small motion cases when computing motion residuals. Note that in [10] the hidden parts of motion layers are extracted too, but in this paper we consider only the visible parts.

It is important to note here that such batch methods require that the motion models have already been estimated for the N considered images before the supports extraction, which implies that a clustering has already been done. Moreover the number N of images processed in the 3D graph can certainly not cover a whole sequence due to complexity issues. These methods also impose matching large numbers of image pairs without exploiting the large amount of implied redundancy.

Complementary to the motion segmentation task, the layer representation and the extracted supports are used to generate as many layer mosaics as extracted motion layers [1,11]. [12] presents not only static mosaics built in batch mode but also dynamic mosaics corresponding to a sequential update of mosaic images. The recurrent problem of global and local realignment in mosaics is addressed in [13].

In this paper we propose a new motion layer extraction system using reference layer mosaics. "*Motion layer*", or simply "*layer*", designates a set of image regions sharing the same motion. The terminology will also be used to simply designate a segmentation label or a layer index.

"*Motion layer mosaic*", or simply "*mosaic*", designates a still image providing a not necessarily complete planar representation integrating the different elements of a given layer observed on a set of images. In our work a mosaic is associated with a reference instant on which it is aligned.

Our motivation is to work with distant images (because motions estimated between distant images are more likely to be different from one region to another than when estimated between consecutive images) while reducing the number

of image pairs to be handled and benefiting from the fact that layer mosaics represent the video sequence in a compact way. Note that mosaics can be reused in region filling or video editing systems, or also for compression tasks.

We present an iterative technique to estimate the motion parameters, to extract the support of each layer, and to create layer mosaics as the segmentation progresses. At the end of the segmentation, like with every segmentation method, it can happen that the obtained boundaries are not precisely the expected ones. Often the user has no other possibility than interacting on every images one after another to correct the unsatisfactory results. We propose an original way for the user to easily provide additional input on the mosaics themselves. This input is then propagated to the whole video interval during a second segmentation pass.

The paper is organized as follows. Section 2 presents our motion layer extraction algorithm and the objective function that we use. Section 3 presents different ways to generate motion layer mosaics, either before or as the segmentation task is conducted. Experimental results are shown in Sect. 4.

2 Motion Segmentation Using Motion Layer Mosaics

According to the assumption usually made in motion segmentation, the scene can be approximated by a set of layers whose motion in the image plane is well described by a low-dimensional parametric model. Additionally, in our work we assume that the number n of layers is known and that the layers keep the same depth order during the whole sequence.

We mainly base our work on [8,9,10]. In our approach, instead of linking temporally many image pairs from the sequence, we propose to link temporally the currently processed image with two sets of motion layer mosaics only. Thus a new image-to-mosaic data term replaces the usual image-to-image motion data term in our objective function. Note that motion layer mosaics are mentioned in the following subsections, although their generation is only presented in Sect. 3.

2.1 Extraction of Motion Layers Using Reference Layer Mosaics

First, the user selects reference instants to divide the video into several time intervals. On reference images, the segmentations and the depth order of the n layers are obtained semi-automatically. We then process sequentially from one time interval $[t_a, t_b]$ to the next. A simplified flow chart of our sequential system after initialization is shown in Fig. 1.

For a given time interval $[t_a, t_b]$ between two reference instants, our aim is to progress from t_a to t_b while obtaining a succession of temporally consistent segmentations. The independent segmentation maps obtained by the user for t_a and t_b may be seen as boundary conditions that are propagated in the interval. In association with these reference instants, as the segmentation progresses, we generate two sets of reference motion layer mosaics $(M_{a,i})_{i\in[0,n-1]}$ and $(M_{b,i})_{i\in[0,n-1]}$.

At current time t, the segmentation of the image I_t is first predicted by projection of the previous segmentation. A dense backward motion field between I_t

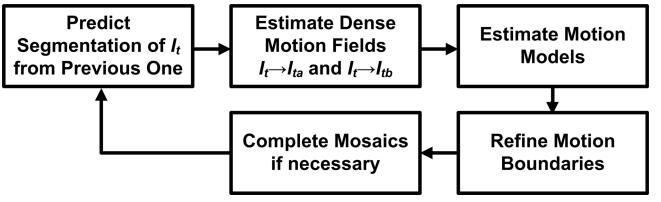

Fig. 1. Simplified flow chart of our algorithm

and I_{t_a} and a dense forward motion field between I_t and I_{t_b} are estimated. A reliability index, based on the Displaced Frame Difference (DFD) is computed for every motion vector of both fields:

$$\forall \alpha \in \{a, b\} \;\; reliab_\alpha(\mathbf{p}) = \max\big(0, 1 - \frac{\|I_t(\mathbf{p}) - I_{t_\alpha}(\mathbf{p} - \mathbf{dp}_\alpha)\|}{\tau_1}\big) \;\; , \tag{1}$$

where $\mathbf{dp}_a$ is the backward motion vector estimated at pixel $\mathbf{p}$, $\mathbf{dp}_b$ the forward one, and τ_1 is an empirical normalization threshold.

As for parametric motion modeling, the affine model $\mathcal{A}$ with 6 degrees of freedom is adopted. Thus for a pixel $\mathbf{p} = (x, y)$ of motion vector $\mathbf{dp}(x, y)$:

$$\mathbf{dp}(x, y) = \mathcal{A}(x, y) = \begin{pmatrix} a_{x0} + a_{xx} \cdot x + a_{xy} \cdot y \\ a_{y0} + a_{yx} \cdot x + a_{yy} \cdot y \end{pmatrix} \;\; . \tag{2}$$

Based on the dense motion fields, forward and backward affine motion models are estimated for each layer of the predicted map. The motion parameters are recovered by a standard linear regression technique but the reliability measure weights the influence of each vector on the estimate. For the motion models, we use the following notations: for layer l, $\mathcal{A}_{a,l}$ is the backward affine model estimated between I_t and I_{t_a} and $\mathcal{A}_{b,l}$ is the forward one estimated between I_t and I_{t_b}.

According to the assumption of brightness constancy, for a pixel $\mathbf{p}$ of the image I_t that belongs to the layer l and that is not occluded in I_{t_a} and I_{t_b}:

$$\forall \alpha \in \{a, b\} \;\; I_t(\mathbf{p}) \approx I_{t_\alpha}\big(\mathbf{p} - \mathcal{A}_{\alpha,l}(\mathbf{p})\big) \;\; . \tag{3}$$

Based on these layered motion models, predicted motion boundaries are finally refined using the graph cut framework.

At this step the layers obtained for I_t are warped into the two reference mosaic sets $(M_{a,i})_{i\in[0,n-1]}$ and $(M_{b,i})_{i\in[0,n-1]}$ using either the forward models or the backward models depending on the mosaic of destination. The stitching of the supports of I_t into the reference mosaics is described in Subsect. 3.3.

The prediction of the segmentation map at next time $t+1$ is done by projection of the segmentation map of I_t using affine motion models estimated from a dense motion field computed between I_t and I_{t+1}. The layers are projected one after another from the most distant to the closest. Since the affine predictions rarely arrive at some integer positions, the same label is assigned to the four nearest

neighbors. This allows attaching a prediction to most of the pixels where no affine prediction arrives. Remaining areas without any predicted label are areas which are occluded in I_t and visible in I_{t+1}.

This is continued for all the images of the interval $[t_a, t_b]$.

2.2 Objective Function

Given the labeling $f = (f_{\mathbf{p}})_{\mathbf{p}\in\mathcal{P}}$ with $f_{\mathbf{p}} \in [0, n-1]$ and $\mathcal{P}$ the pixel set to be segmented, we consider the following objective function, which is the sum of two standard terms (color data term and spatial smoothness) described in [14], with an original image-to-mosaic data term and a temporal term:

$$E(f) = \lambda_0 \underbrace{\sum_{\mathbf{p}\in\mathcal{P}} C_{\mathbf{p}}(f_{\mathbf{p}})}_{E_{\text{color}}(f)} + \lambda_1 \underbrace{\sum_{(\mathbf{p},\mathbf{q})\in\mathcal{C}} V_{\mathbf{p},\mathbf{q}}(f_{\mathbf{p}}, f_{\mathbf{q}})}_{E_{\text{smooth}}(f)} + \lambda_2 \underbrace{\sum_{\mathbf{p}\in\mathcal{P}} D_{\mathbf{p}}(f_{\mathbf{p}})}_{E_{\text{mosaic}}(f)} + \lambda_3 \underbrace{\sum_{\mathbf{p}\in\mathcal{P}} \Psi_{\mathbf{p}}(f_{\mathbf{p}})}_{E_{\text{temp}}(f)} \quad , \quad (4)$$

where $\mathcal{C}$ is the set of neighbor pairs with respect to 8-connectivity, and $(\lambda_i)_{i\in[0,3]}$ are positive parameters that weight the influence of each term.

$C_{\mathbf{p}}(f_{\mathbf{p}})$ is a standard color data penalty term at pixel $\mathbf{p}$, set as negative log-likelihood of color distribution of the layer $f_{\mathbf{p}}$ [8,15]. This distribution consists of a Gaussian Mixture Model (GMM) computed on the mosaics $M_{a,f_{\mathbf{p}}}$ and $M_{b,f_{\mathbf{p}}}$ before that the segmentation process starts (see Sect. 3 for details on mosaic generation).

$V_{\mathbf{p},\mathbf{q}}(f_{\mathbf{p}}, f_{\mathbf{q}})$ is a standard contrast-sensitive regularization term:

$$V_{\mathbf{p},\mathbf{q}}(f_{\mathbf{p}}, f_{\mathbf{q}}) = \frac{1}{\text{dist}(\mathbf{p},\mathbf{q})} \exp\left(-\frac{\|I_t(\mathbf{p}) - I_t(\mathbf{q})\|^2}{2\sigma^2}\right) \cdot T(f_{\mathbf{p}} \neq f_{\mathbf{q}}) \quad , \quad (5)$$

where σ is the standard deviation of the norm of the gradient, $T(.)$ is 1 if the argument predicate is true and 0 otherwise and dist(.) is the euclidean distance.

$D_{\mathbf{p}}(f_{\mathbf{p}})$ is a new image-to-mosaic data penalty term at pixel $\mathbf{p}$ for the motion model corresponding to layer $f_{\mathbf{p}}$. It corresponds to a residual computed between the image and the two mosaics for the concerned layer. These mosaics contain more information than the image itself since it is an accumulation of all the elements that have been visible in the previously segmented images, that could have then disappeared but also that could reappear. Thus, with a unique image-to-mosaic matching we can do as well as with numerous image-to-image comparisons. $D_{\mathbf{p}}(f_{\mathbf{p}})$ is defined as:

$$D_{\mathbf{p}}(f_{\mathbf{p}}) = \min_{\alpha\in\{a,b\}} \left(r_\alpha(\mathbf{p}, f_{\mathbf{p}})\right) \quad , \quad (6)$$

$$r_\alpha(\mathbf{p}, f_{\mathbf{p}}) = \begin{cases} \arctan\left(\left\|I_t(\mathbf{p}) - M_{\alpha,f_{\mathbf{p}}}(\mathbf{p}'_\alpha)\right\|^2 - \tau_2\right) + \frac{\pi}{2} \\ \qquad\qquad \text{if } \mathbf{p}'_\alpha \text{ belongs to the mosaic support} \\ +\beta \qquad\qquad \text{otherwise} \end{cases} \quad , \quad (7)$$

where $\mathbf{p}'_\alpha$ is the position associated in the mosaic $M_{\alpha,f_{\mathbf{p}}}$ to $\mathbf{p}$ in I_t according to $\mathcal{A}_{\alpha,f_p}$. This smooth penalty and its threshold parameter τ_2 allow a soft distinction between low residuals (well classified pixels) and high residuals (wrongly

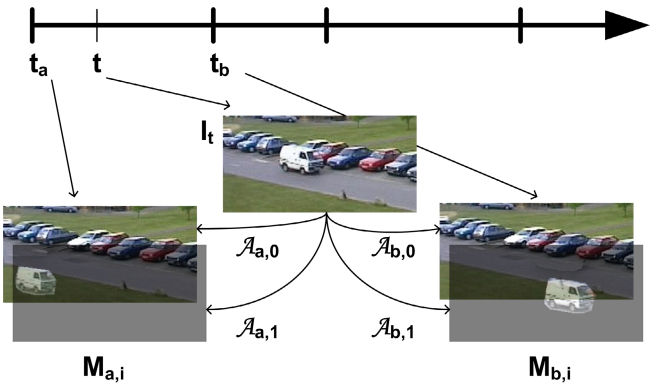

Fig. 2. Matching between the current image to be segmented and the mosaics of the two layers 0 and 1 that respectively correspond to the background and to the moving van at reference instants t_a and t_b.

classified pixels or occluded pixels). For all our experiments, $\tau_2 = 50$ and $\beta = 10$. Figure 2 illustrates this matching between the current image to be segmented and the two sets of reference mosaics.

The last temporal term of the energy function enforces temporal consistency of labeling between consecutive images. It is defined as:

$$\Psi_{\mathbf{p}}(f_{\mathbf{p}}) = \begin{cases} 1 \text{ if } f_{\mathbf{p}} \neq f'_{\mathbf{p}} \text{ and } f'_{\mathbf{p}} \neq \emptyset \\ 0 \text{ otherwise} \end{cases}, \tag{8}$$

where $f'_{\mathbf{p}}$ is the predicted label of pixel $\mathbf{p}$ at instant t, and $\emptyset$ is the blank label for the appearing areas without any predicted label.

2.3 Minimization

The global energy (4) is minimized using graph-cuts [16,17]. As we want to handle an arbitrary number of layers, we use the α-expansion algorithm [18] to solve the multi-labels problem. We obtain simultaneously all the layer supports for the considered instant. Moreover our algorithm provides dense segmentation maps without any additional label for noise, occluded pixels or indetermination (contrary to [9,10]).

Given that we work at the pixel level, building a graph on the whole image and conducting minimization on it is computationally expensive in the context of video analysis and editing. Moreover, because we assume temporal consistency on the segmentation maps, it is not useful to question all of the pixels at every instant. Consequently we propose to build the graph on an uncertainty strip around the predicted motion boundaries and on appearing areas only. Note that appearing areas have no predicted label, so they are also considered as uncertain. Pixels ignored by the graph retain their predicted label that we assume correct. Such a graph restriction constrains the segmentations both spatially and temporally.

Pixels on the outlines of the graph area keep their predicted label and constitute boundary conditions for the segmentation to be obtained. These labeled pixels are considered as seeds and provide hard constraints which are satisfied setting to specific values the weights of the links which connect them to the terminals (see [14]).

3 Generation of Motion Layer Mosaics

The proposed motion segmentation system requires for each reference instant as many mosaics as there are layers in the scene. But how to provide such mosaics as input for our system whereas layer supports are needed to generate these mosaics? We describe three ways to obtain these motion layer mosaics. For the generation of each mosaic the user decides which way to be employed regarding the ease of use.

3.1 Simple Case

The first approach concerns the simplest particular case: the one of a foreground object fully visible in the image associated to the chosen reference instant. The interest of such a situation can serve as a guide in the manual choice of reference instants.

For such a foreground object the mosaic is made up of the region corresponding to the complete object. The whole remaining area of the mosaic is empty. Since the considered object is fully visible at the reference instant, the mosaic does not need to be further completed. Thus the user can easily provide the mosaic before the motion segmentation step. The object boundaries can be obtained by a semi-automatic tool for still image segmentation like [14,15]. An example of this simple case is illustrated with the white van in Fig. 3.

Fig. 3. Reference mosaics for PETS 2001 sequence. First row, background mosaics obtained by semi-automatic stitching from the two reference images. Second row, foreground mosaics obtained by interactive segmentation in the two still reference images.

3.2 Semi-automatic Stitching

The second approach is inspired by the stitching tools commonly used to generate panoramas from several photographs [13,19]. Such tools usually detect feature points in each photograph of the set. Then for each pair of images with a common overlapping area, the features belonging to this area are matched to compute the global transformation between the two photographs.

In our case, the set of photographs is replaced by two distant images from the original sequence. The user is guided by his/her knowledge of the sequence to choose adequately the two images such that they will provide together the whole layer support.

While one assumes that there is only one layer in the common cases of panorama, in our case we have to remember that there are several layers. The user semi-automatically segments both distant reference images and indicates the layer correspondences. Then one transformation per layer is estimated. Instead of automatic features matching, the user can manually select pairs of control points.

This approach is particularly well adapted for the background if the areas occluded by foreground objects in the first image are totally disoccluded in the last image. This example is illustrated by the background layer in Fig. 3.

3.3 Joint Mosaic Generation and Motion Segmentation

The third approach, which can always be applied, does not require any user interaction. The generation of the mosaics is embedded in our time-sequential segmentation system. For each layer, the supports are stitched automatically together as they are extracted. The first support results from the semi-automatic segmentation of the reference image.

Motion layer extraction consists in extracting the layer supports but also in estimating the motion model parameters of each layer. Thus, using the motion models, layers are warped into the corresponding reference mosaics we want to build. This means that in this case reference mosaics evolve as the segmentation process progresses unlike the mosaics built in the two previous approaches.

Regions that appear for the first time are copied into the mosaics. For the regions already seen, we simply keep the values of the first apparition without any refreshment to preserve the reference data in case of classification errors. Figure 7 shows mosaics obtained this way using 20 images.

Given the notations introduced in Sect. 2:

- Either the pixel $\mathbf{p}$ of layer $f_{\mathbf{p}}$ in I_t has already been disoccluded in the images previously segmented, in which case the information is already available in the reference mosaics $M_{a,f_{\mathbf{p}}}$ and $M_{b,f_{\mathbf{p}}}$ generated for layer $f_{\mathbf{p}}$.
- Or the pixel $\mathbf{p}$ of layer $f_{\mathbf{p}}$ in I_t occurs for the first time, in which case the information is missing in the reference mosaics, so we copy it:

$$\forall \alpha \in \{a, b\} \;\; M_{\alpha,f_{\mathbf{p}}}\big(\mathbf{p} - \mathcal{A}_{\alpha,f_{\mathbf{p}}}(\mathbf{p})\big) = I_t(\mathbf{p}) \;\; . \tag{9}$$

Table 1. Summary of steps and degree of automaticity

reference instants selection	by the user (or automatic if periodic)
reference images segmentation	semi-automatic
motion estimation and segmentation	automatic
mosaic generation	
Subsect. 3.1	cf. reference images segmentation
Subsect. 3.2	semi-automatic
Subsect. 3.3	automatic
mosaic correction (if required)	manual

In case of segmentation mistakes, the mosaics will be directly affected. In other words, misclassified areas may be stitched into the wrong mosaic. We voluntarily do not try to conceal such errors during mosaic generation. The user is allowed to efficiently remove them at the end of the first segmentation pass. A second segmentation pass with such corrected mosaics as input can improve notably the accuracy of the motion boundaries (see Subsect. 4.3).

Table 1 summarizes the steps of our system and their degree of automaticity.

4 Experimental Results

4.1 Validation of Our Image-to-Mosaic Energy Term

First we tested our algorithm on a video surveillance sequence from the PETS 2001 database (courtesy of The University of Reading, UK). The camera is fixed. The scene represents a parking with a single moving object: a white van. We chose to process only one interval of 60 images because the motion of the van is more complex than a planar motion in the rest of the sequence.

The van is fully visible in every image of the processed interval. Mosaics were easily obtained before we launched the motion segmentation task. This simple case allowed us to focus on motion layer extraction and not on mosaic generation. Thus we validated our image-to-mosaic energy term by imposing null weights to the color data term and to the temporal one.

We present the reference mosaics we used in Fig. 3. The algorithm extracted well both layers (Fig. 4). The only errors occurred when the moving white van goes in front of the parked white car. Our image-to-mosaic term is efficient except in these "white on white" regions.

Note that through the windshield of the van, pixels are considered as belonging to the background, which is interesting for some applications. If the white van were to be removed by region filling, the windshield should be classified as foreground. This can be obtained by incorporating the temporal term (see Fig. 5).

4.2 Results on *Flower Garden* Sequence

We tested our algorithm on the first 20 images of the well known *Flower Garden* sequence. 0ur results shown in Fig. 6 are as good as the best ones in the literature.

Fig. 4. Results on PETS 2001 sequence without temporal constraints. Motion layer boundaries are superimposed on the images. Note the temporal inconsistencies on the windshield of the van, due to its transparency. $\lambda_0 = 0$, $\lambda_1 = 30$, $\lambda_2 = 17$ and $\lambda_3 = 0$.

Fig. 5. Results on PETS 2001 sequence with temporal constraints. Motion layer boundaries are superimposed on the images. The windshield of the van is always classified as foreground. $\lambda_0 = 0$, $\lambda_1 = 30$, $\lambda_2 = 17$ and $\lambda_3 = 15$.

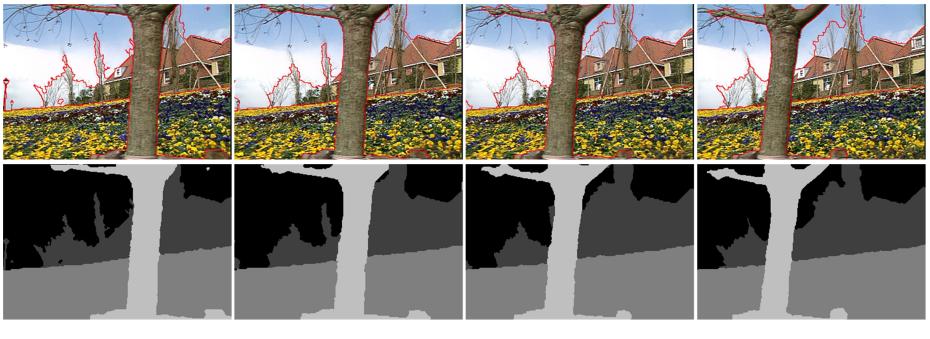

Fig. 6. Results on *Flower Garden* with simultaneous automatic mosaics generation. Four images with superimposed motion boundaries and corresponding segmentation maps. Depth display convention: the darker is the region, the more distant is the layer. Please report to [8,9] for results comparisons. $\lambda_0 = 1$, $\lambda_1 = 11$, $\lambda_2 = 17$ and $\lambda_3 = 3$.

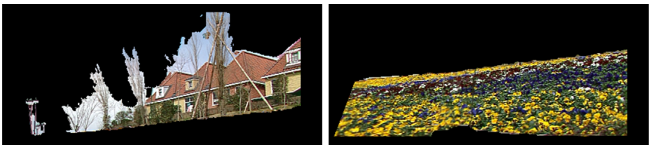

Fig. 7. Layer mosaics automatically generated from 20 different images of *Flower Garden* sequence. Please report to [1] for "flowerbed" mosaic comparison.

The precise extraction of the branches would require a matting method. The end of the sequence was satisfactorily processed too in a second interval but results are not shown due to space constraints.

Figure 7 shows two of the mosaics generated automatically from the 20 images of the interval as the motion segmentation progressed. Compared with the "flowerbed" accumulation of [1], there is no insertion of parts of "houses" layer in our "flowerbed" mosaic and the quality of the stitching is as good without any temporal median operation.

4.3 Results on *Carmap* Sequence

The *Carmap* sequence is also tested in [9,10]. Because of 3D motion present in the sequence we divided it into three time intervals to make the problem easier. Segmentation was done on the three intervals. Due to space constraints, results on the first two intervals are not supplied.

A first experiment was done building the mosaics as the segmentation was performed. Results are shown in Fig. 8. The approximation of the motion of the car by an affine model failed because of the out-of-plane 3D rotation and of the importance of the time distance between matched images (relative to the amplitude of apparent motion in the scene). Instead of subdividing the interval into two shorter ones and starting all over again, we added some modest user input on the generated mosaics before launching a second segmentation pass.

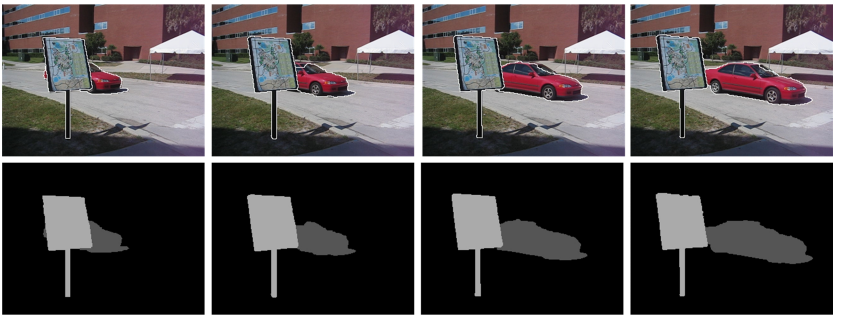

Fig. 8. Results on *Carmap* sequence after first pass. Note the misclassifications of the windshield. $\lambda_0 = 1$, $\lambda_1 = 30$, $\lambda_2 = 14$ and $\lambda_3 = 10$.

Fig. 9. Mosaic of the background. From left to right: obtained mosaic after first pass, detail of the white square that wrongly includes some parts of the car, misclassification removal by the user. In black, remaining holes in the mosaic: these regions are never visible in the processed interval.

Fig. 10. Results on *Carmap* sequence after light weight mosaic correction by the user and second segmentation pass. Misclassifications of the windshield are fixed. $\lambda_0 = 1$, $\lambda_1 = 30$, $\lambda_2 = 14$ and $\lambda_3 = 10$.

Figure 9 shows how the user erased misclassified regions in the background mosaic using a standard image editing rubber. A second segmentation pass was performed with this modified mosaic as input mosaic for the background layer. The interest to work with layer mosaics is here fully demonstrated. Corrections applied by the user in the still mosaic behave as seeds which are propagated to all the images during the second segmentation pass. Results are presented in Fig. 10. After modest user interaction on one mosaic only, the second segmentation pass provided results at least as good as the ones of batch methods [9,10].

4.4 Discussion

Because we do not have any ground-truth, the quality of the presented results has unfortunately to be evaluated subjectively. In our opinion we obtained results of at least similar quality to those of simultaneous batch approaches that are more expensive. However, remember that the manual intervention is integral to the present approach, so comparison to more automatic approaches requires precautions. Our contribution is well adapted for post-production applications for example. In such a context, user interaction is relevant. The constraint of perfect visual quality of the resulting sequence is so high that it is generally preferable that the approach relies on some user input even if this reduces the degree of automation.

Let us discuss for example the initialization step which is crucial in motion layer extraction methods, whether they are automatic or semi-automatic. The automatic initialization proposed in [9] uses the N first images of the sequence

and is based on time-consuming steps like seed regions expansion and region merging. First, the user could quickly obtain similar initialization with modest effort, as we do. Second, it may happen that two layers have different motions at the end of the sequence only. Such an automatic initialization will certainly merge them with no way to separate them efficiently later, whereas a semi-automatic initialization allows the user to distinguish both layers.

5 Conclusion

In this paper we present a new motion layer extraction method based on layer mosaics. For some reference instants of the sequence, we generate motion layer mosaics in which we accumulate the information of each layer. This mosaic generation is done either initially or as the segmentation process progresses. We propose to exploit this compact layered representation of the sequence in three ways during the segmentation process.

First, we reduce the number of image pairs to be handled. Motion is estimated between each image and two of the reference instants only, and not for large numbers of image pairs as with the simultaneous batch approaches.

Second, since the previous point implies that we work with distant images, we benefit from the fact that motions estimated between distant images are more likely to be different from one region to another than when estimated between consecutive images. This helps the segmentation process.

Third, if after a first segmentation pass, results are not satisfactory, the user can easily add modest input directly on the generated mosaics before starting a second segmentation pass during which input is propagated on the whole video interval.

Promising results show the potential of our method. In future work, we will integrate in our algorithm a matting step to be able to handle sequences with significant semi-transparent regions.

References

1. Wang, J., Adelson, E.: Representing moving images with layers. IEEE Trans. on Image Processing 3(5), 625–638 (1994)
2. Ayer, S., Sawhney, H.S.: Layered representation of motion video using robust maximum-likelihood estimation of mixture models and MDL encoding. In: ICCV, pp. 777–784 (1995)
3. Black, M.J., Anandan, P.: The robust estimation of multiple motions: Parametric and piecewise-smooth flow fields. Computer Vision and Image Understanding 63(1), 75–104 (1996)
4. Mémin, E., Pérez, P.: Hierarchical estimation and segmentation of dense motion fields. IJCV 46(2), 129–155 (2002)
5. Bouthemy, P., François, E.: Motion segmentation and qualitative dynamic scene analysis from an image sequence. IJCV 10(2), 157–182 (1993)
6. Cremers, D., Soatto, S.: Motion competition: A variational approach to piecewise parametric motion segmentation. IJCV 62(3), 249–265 (2005)

7. Odobez, J.M., Bouthemy, P.: Direct incremental model-based image motion segmentation for video analysis. Signal Processing 66(2), 143–155 (1998)
8. Dupont, R., Paragios, N., Keriven, R., Fuchs, P.: Extraction of layers of similar motion through combinatorial techniques. In: Rangarajan, A., Vemuri, B.C., Yuille, A.L. (eds.) EMMCVPR 2005. LNCS, vol. 3757, pp. 220–234. Springer, Heidelberg (2005)
9. Xiao, J., Shah, M.: Motion layer extraction in the presence of occlusion using graph cuts. PAMI 27(10), 1644–1659 (2005)
10. Dupont, R., Juan, O., Keriven, R.: Robust segmentation of hidden layers in video sequences. Technical report, CERTIS - ENPC (January 2006)
11. Min, C., Yu, Q., Medioni, G.: Multi-layer mosaics in the presence of motion and depth effects. In: ICPR, pp. 992–995 (2006)
12. Irani, M., Anandan, P., Bergen, J., Kumar, R., Hsu, S.: Efficient representations of video sequences and their applications. Signal Processing: Image Communication 8(4), 327–351 (1996)
13. Shum, H.Y., Szeliski, R.: Construction and refinement of panoramic mosaics with global and local alignment. In: ICCV (1998)
14. Boykov, Y., Jolly, M.P.: Interactive graph cuts for optimal boundary & region segmentation of objects in n-d images. In: ICCV (2001)
15. Rother, C., Kolmogorov, V., Blake, A.: "Grabcut": interactive foreground extraction using iterated graph cuts. ACM Trans. Graph. 23(3), 309–314 (2004)
16. Boykov, Y., Kolmogorov, V.: An experimental comparison of min-cut/max-flow algorithms for energy minimization in vision. PAMI 26(9), 1124–1137 (2004)
17. Kolmogorov, V., Zabih, R.: What energy functions can be minimized via graph cuts? PAMI 26(2), 147–159 (2004)
18. Boykov, Y., Veksler, O., Zabih, R.: Fast approximate energy minimization via graph cuts. PAMI 23(11), 1222–1239 (2001)
19. Brown, M., Lowe, D.: Recognising panoramas. In: ICCV, pp. 1218–1225 (2003)