

# TIME-SEQUENTIAL EXTRACTION OF MOTION LAYERS

Matthieu Fradet, Patrick Pérez\* and Philippe Robert

Thomson Corporate Research, Rennes, France, \*INRIA, Rennes - Bretagne Atlantique, France

## ABSTRACT

A new time-sequential approach for motion layer extraction is presented. We assume that the scene can be described by a set of layers associated to affine motion models. In one or more key frames, the segmentation is obtained using a semi-automatic method. At a subsequent instant, the first step of the proposed algorithm is the prediction of the segmentation from one image to the next one, using motion models estimated for each layer. The second step is the refinement of the predicted motion boundaries by graph cut. Only the appearing areas and a strip around the predicted boundaries are questioned. A new rigidity constraint improves the temporal consistency of foreground rigid objects. Experimental results show that our sequential approach is at least as effective as more complex simultaneous approaches while being less computationally demanding.

**Index Terms**— Motion Analysis, Video Segmentation, Motion Layers, Motion Boundaries, Graph Cut.

## 1. INTRODUCTION

Despite its long history, the problem of segmenting video in regions of similar motion is still a very active research topic in computer vision. 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 motions in the image plane are 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.

In [1] a dense motion field previously estimated is segmented. Parametric models are iteratively *a)* estimated on rough regions, *b)* used to refine the regions and *c)* finally updated according to the new supports. Once convergence is obtained, segmentation map is predicted for the next image, new motion models are estimated, etc.

Some authors [2, 3] propose to combine dense motion estimation and parametric segmentation, while some others extract layers either jointly [4], or one after another [5] using a direct parametric segmentation with no optical flow computation.

More recently, video segmentation was formulated into the graph cut framework. Sequential approaches such as [6] provide a segmentation map for the current image taking into account the previous labeling only. But the whole segmentation is questioned again in the subsequent image, at the expense of temporal consistency.

Some researchers moved naturally to simultaneous batch approaches [7, 8] to increase temporal consistency. To do so, they use 3D graphs at the pixel level that allow the simultaneous segmentation of  $N$  images. Such methods remove certain artifacts that successive 2D graph optimizations can create. ([8] presents a method to extract

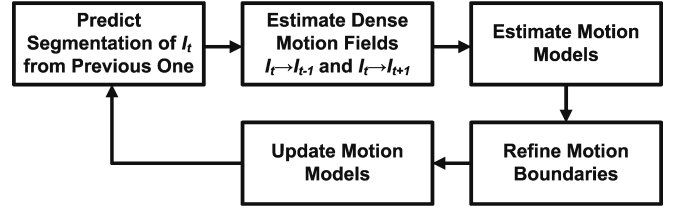


Fig. 1. Simplified Flow Chart of our Algorithm.

the hidden parts of motion layers too, but in this paper we consider only the visible parts.)

The main limitation of simultaneous approaches is their computational complexity. The number  $N$  of processed images can certainly not cover a whole sequence due to complexity issues. Also, assuming all pixel labels unknown within the temporal window of interest does not allow any restriction of the area on which the graph is built.

In contrast, we introduce a time-sequential approach which, with modest user input in one or few frames, provides results at least as good as those obtained by latter batch approaches at a lower computational cost.

In our work, we assume that the layers keep the same depth order during the whole sequence. On the first image, as well as on few subsequent images if necessary, the segmentation is obtained using an interactive graph cut-based method that exploits both motion and color information.

At current time  $t$ , the segmentation of image  $I_t$  is first predicted by projection of the previous segmentation. A dense forward motion field between  $I_t$  and  $I_{t+1}$  and a backward motion field between  $I_t$  and  $I_{t-1}$  are estimated. Based on them, forward and backward affine motion models are estimated for each layer according to the predicted segmentation. Predicted motion boundaries are finally refined using graph cut to minimize an energy composed of motion, color, spatial smoothness, temporal terms and a new rigidity constraint term. This is continued for all the images of the sequence.

A simplified flow chart of our sequential system after initialization is shown in Figure 1.

The paper is organized as follows. Section 2 addresses user interaction. Section 3 presents the graph restriction and our motion boundaries propagation and refinement algorithm. Experimental results are shown in Section 4.

## 2. INTERACTIVE SEGMENTATION OF ONE IMAGE

This step is systematically required for the first frame, but can also be used later in the sequence for re-initialization if needed.

For the considered image  $I_t$ , the user provides some large and loose seeds to mark the  $n$  different layers. These seeds are generally



**Fig. 2.** Example of polygonal seeds given by the user. Left to right: original image, seeds, obtained segmentation in 5 layers. Depth display convention: the darker is the seed, the more distant is the layer.

polygonal regions, ordered by depth. An example of such seeds is given in Figure 2.

The seeds are used as layer supports to compute Gaussian Mixture Models (GMMs) in the RGB space. They are also used jointly with a forward dense motion field to estimate one affine motion model per layer.

Given the labeling  $f = (f_p)_{p \in \mathcal{P}}$  with  $f_p \in [0, n-1]$  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 [9], and of the motion-based term described in [7]:

$$E(f) = \underbrace{\sum_{p \in \mathcal{P}} C_p(f_p)}_{E_{\text{color}}(f)} + \lambda_1 \underbrace{\sum_{(p,q) \in \mathcal{C}} V_{p,q}(f_p, f_q)}_{E_{\text{smooth}}(f)} + \lambda_2 \underbrace{\sum_{p \in \mathcal{P}} D_p(f_p)}_{E_{\text{motion}}(f)} \quad (1)$$

$$D_p(f_p) = \arctan(\|I_t(\mathbf{p}) - I_{t+1}(\mathbf{p}')\|^2 - \tau_1) + \frac{\pi}{2} \quad (2)$$

where  $\mathcal{C}$  is the set of neighbor pairs with respect to 8-connectivity, and  $\lambda_1$  and  $\lambda_2$  are positive parameters which weight the influence of each term.

$C_p(f_p)$  is a standard color data penalty term at pixel  $\mathbf{p}$ , set as the negative log-likelihood of color distribution of the layer  $f_p$ . This distribution consists of the GMM computed on the seeds of the layer  $f_p$ .

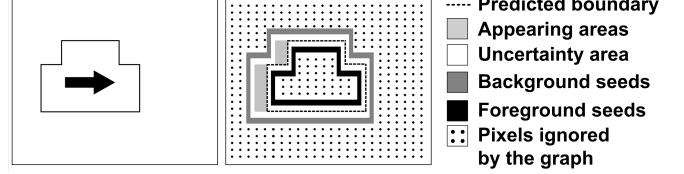
$V_{p,q}(f_p, f_q)$  is a standard contrast-sensitive regularization term.

$D_p(f_p)$  is a data penalty term at pixel  $\mathbf{p}$  for the motion model corresponding to layer  $f_p$  ( $\mathbf{p}'$  is the correspondent in  $I_{t+1}$  of  $\mathbf{p}$  in  $I_t$  according to this parametric motion model). This smooth penalty and its threshold parameter  $\tau_1$  allow a soft distinction between low residuals (well classified pixels) and high residuals (wrongly classified pixels or occluded pixels). In our experiments we chose  $\tau_1 = 50$ . Adding this motion term solves more easily ambiguities due to colors, with no need for the user to introduce additional seeds. If two different objects have similar colors but different motions, an interactive method based on colors only would force the user to provide additional seeds in such regions.

### 3. MOTION BOUNDARIES REFINEMENT

#### 3.1. Graph Restriction

In our sequential approach, we propose to predict the labeling, and hence motion boundaries, from the previous instant and to consider, around these predicted boundaries, an uncertainty strip in which labeling can be modified. Notice that appearing areas have no predicted labels. That is why they are also considered as uncertain. In the other areas, we assume that the predicted labels are correct and associated pixels are ignored by the graph. Such an assumption reduces significantly the size of the graph and constrains the segmentation both spatially and temporally.



**Fig. 3.** Graph Restriction. Case of a moving foreground object and a stationary background. (Left, previous segmentation already obtained.)

Figure 3 illustrates this graph restriction in a binary case. The width  $w$  of the uncertainty strip is defined by the user. In our experiments we chose  $w = 12$ . It includes 5 pixels on each side of the boundaries, and 1 more pixel on each side to have boundary conditions. Pixels belonging to these 1-pixel strips are considered as seeds and provide hard constraints which are satisfied by setting to specific values the weights of the links that connect these seeds with the terminals (see [9]).

This way to build the graph is particularly well adapted to our motion boundaries refinement.

#### 3.2. Objective Function

Moreover, to increase the effect of temporal consistency, not only the currently processed image and the previous one but also the next image are taken into account, to form a triplet. Forward and backward motion fields are consequently estimated. We compare these two fields to improve accuracy of the estimated vectors in the occlusion areas. A reliability index, based on the Displaced Frame Difference (DFD) is computed for every motion vector of both fields:

$$r_\alpha(\mathbf{p}) = \max(0, 1 - \frac{\|I_t(\mathbf{p}) - I_{t+\alpha}(\mathbf{p} - \mathbf{dp}_\alpha)\|}{\tau_2}), \alpha \in \{-1, +1\} \quad (3)$$

where  $\mathbf{dp}_\alpha$  is the motion vector, either forward or backward, estimated at pixel  $\mathbf{p}$  and  $\tau_2$  is an empirical normalization threshold.

Then for a pixel whose forward vector has a lower reliability than its backward vector, we correct the forward field by replacing the forward vector with the opposite of the backward one. The same correction is done for the backward field. This correction improves the estimated motion fields before the motion model approximation step. At the end of this motion estimation step, two corrected motion fields (forward and backward) and their associated pixel-wise reliability maps are available.

The motion model parameters of each layer are recovered by a standard linear regression technique but we use the reliability measure to weight the influence of each vector on the model.

The energy to be minimized at current instant extends (1) as follows:

- The set of pixels  $\mathcal{P}$  is only a fraction of the original pixel grid, as explained in 3.1.
- We keep the same expressions for color data and spatial smoothness terms.
- We adapt the motion data term to our triplet-based sequential setup. Thus, (2) becomes

$$D_p(f_p) = \min_{\alpha \in \{-1, +1\}} (\arctan(\|I_t(\mathbf{p}) - I_{t+\alpha}(\mathbf{p}'_\alpha)\|^2 - \tau_1) + \frac{\pi}{2}) \quad (4)$$

where  $\mathbf{p}'_\alpha$  is the correspondent in  $I_{t+\alpha}$  of  $\mathbf{p}$  in  $I_t$  according to the affine motion model of the layer  $f_p$ .

- Like in [6, 7, 8], we introduce a fourth term to enforce temporal constraints. It maintains the temporal consistency of the segmentation between current frame  $I_t$  and frame  $I_{t-1}$  already segmented. Thanks to the estimated motion models, we define the temporal term as follows:

$$E_{temp}(f) = \sum_{\mathbf{p} \in \mathcal{P}} \psi(\mathbf{p}) \quad (5)$$

$$\psi(\mathbf{p}) = \begin{cases} 1 & \text{if } f_{\mathbf{p}} \neq f_{\mathbf{p}'} \text{ and } f_{\mathbf{p}}' \neq \emptyset \\ 0 & \text{otherwise} \end{cases} \quad (6)$$

where  $\mathbf{p}'$  is the correspondent of  $\mathbf{p}$  in  $I_{t-1}$  according to the backward model of the layer  $f_{\mathbf{p}}$ ,  $f_{\mathbf{p}}'$  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.

- We add, when appropriate, a new rigidity constraint to increase temporal consistency of foreground layers that remain non-occluded and to avoid, this way, the insertion of the corresponding labels in the appearing areas:

$$E_{rigid}(f) = \sum_{\mathbf{p} \in \mathcal{P}} \phi(\mathbf{p}) \quad (7)$$

$$\phi(\mathbf{p}) = \begin{cases} 1 & \text{if } f_{\mathbf{p}} \in \mathcal{S}_{rigid} \text{ and } f_{\mathbf{p}}' = \emptyset \\ 0 & \text{otherwise} \end{cases} \quad (8)$$

where  $\mathcal{S}_{rigid}$  is the set of labels corresponding to the layers on which the rigidity constraint is applied.

The global energy to be minimized at every instant is:

$$E(f) = E_{color}(f) + \lambda_1 E_{smooth}(f) + \lambda_2 E_{motion}(f) + \lambda_3 E_{temp}(f) + \lambda_4 E_{rigid}(f) \quad (9)$$

where  $\lambda_1, \lambda_2, \lambda_3$  and  $\lambda_4$  are positive parameters which weight the influence of each term. They are set by the user and depend on the reliability of the motion estimation and on the characteristics of the sequence. Energies (1) and (9) are minimized using graph cuts [10, 11]. As we want to handle an arbitrary number of layers, we use the  $\alpha$ -expansion algorithm [12] to solve the multi-labels problem.

#### 4. EXPERIMENTAL RESULTS

The proposed algorithm was tested on the same three sequences that are shown in [6, 7, 8]. Our algorithm provides dense segmentation maps without any additional label for noise, occluded pixels or indetermination (contrary to [7, 8]).

Figure 4 shows our results on the *Calendar* sequence. Only one additional user interaction on frame 40 was required. The obtained segmentations are correct even at the end of the sequence, which is usually not processed, where the ball and the train regions are not in contact anymore. Thanks to our rigidity constraint the ball support never stretches over similar color regions of the background.

Figure 5 shows the satisfactory results obtained for the *Flowers* sequence even when the user only provides a segmentation map for the first image. Branches structures are really thin and their precise extraction would require a matting method.

Concerning the *Carmap* sequence, the main difficulty is the large occlusion of the car whose front is hidden by the foreground map at the beginning, and whose wheels and shadow are black like the map edges. Moreover the motions of the map (notably the stem) and of the background are almost similar.

Figure 6 shows the results obtained when the user provided rough segmentation for the images 1 and 11, with and without rigidity constraint on the map layer. The simultaneous approaches seem to encounter the same difficulty to classify as “car” the front of the car as soon as it appears to the right of the map. The contribution of the rigidity constraint is noticeable: it avoids classification mistakes in uniform dark areas (wheels, shadow of the car and edges of the map) where neither the color term nor the motion term are sufficiently discriminative. Notice that even the stem of the map is well segmented.

sequence	image size	mean graph size	$n$	CPU/frame
<i>Calendar</i>	352 x 240	11647 pix.	4	$\sim 6$ s.
<i>Flowers</i>	352 x 240	22886 pix.	4	$\sim 6$ s.
<i>Carmap</i>	320 x 240	9192 pix.	3	$\sim 3$ s.

**Table 1.** Mean CPU times including all processings. ( $n$  is the number of layers).

All the experiments were performed on a P-4 3.6GHz machine. Including all processings (motion estimation, GMMs computation, graph construction, labeling . . .), the time required for the segmentation of one image mainly depends on the number of layers (see Table 1). In [7], the segmentation of one image is less than 30 seconds in average on a P-4 2.0GHz machine.

#### 5. CONCLUSION

We presented a new sequential algorithm for motion layer extraction. We propose to predict motion boundaries from one image to the next one and to refine them by graph cut. Only the appearing areas and a strip around the predicted boundaries are questioned. Results show that our sequential method is less expensive than complex simultaneous methods while providing results of at least similar quality. It includes an original rigidity constraint whose usefulness was demonstrated. Moreover the user can interrupt the system as soon as a segmentation map is not satisfactory, without having to wait that the whole sequence is processed. However it turns out that such interactions are rarely required.

The interactive initialization step could be avoided using an automatic process to estimate the number of layers and their initial parameters (e.g. split and merge process repeated until stability). But note that if the initial segmentation is not accurate enough or if the estimated number of layers is not appropriate, our algorithm will not insure a fast convergence to expected segmentations.

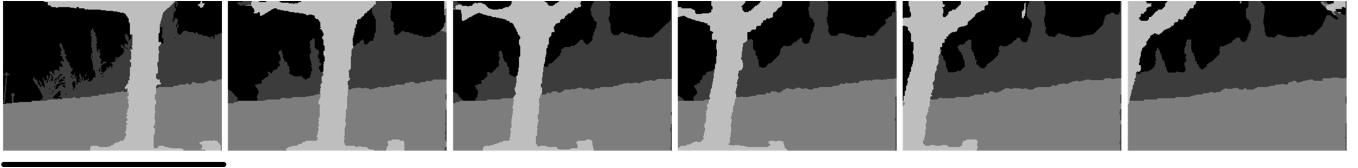
In future work, we will integrate in our algorithm a step of generation of layer mosaic to store the disappearing areas. Such progressively built mosaics could help the segmentation of subsequent images, notably in cases where disappeared areas reappear later in the sequence after long occlusions.

#### 6. REFERENCES

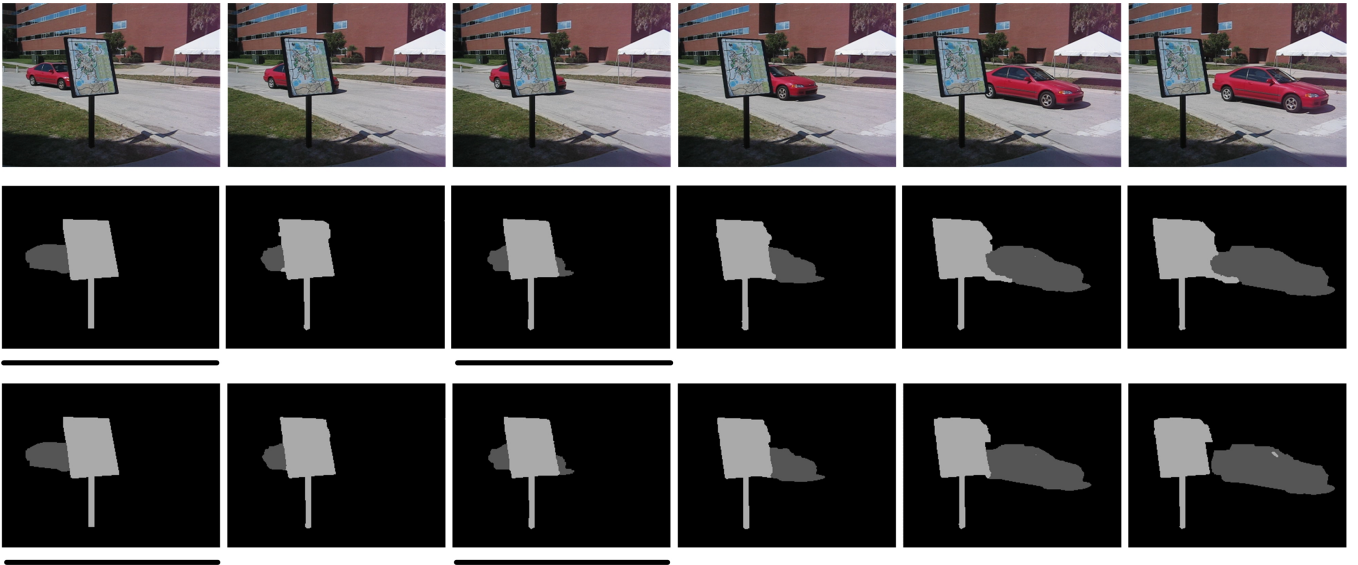
- [1] J. Wang and E. Adelson, “Representing moving images with layers,” *IEEE Trans. on Image Processing*, vol. 3, no. 5, pp. 625–638, September 1994.
- [2] M. J. Black and P. Anandan, “The robust estimation of multiple motions: Parametric and piecewise-smooth flow fields,” *Computer Vision and Image Understanding*, vol. 63, no. 1, pp. 75–104, 1996.



**Fig. 4.** Results on *Calendar* sequence. Frames 1, 39, 40, 70, 84, 99 are shown. Underlined segmentations were obtained with user interaction. Original images of this standard sequence are not displayed due to space constraints. Please report to [6, 7] for them, and for results comparisons.



**Fig. 5.** Results on *Flowers* sequence. Frames 1, 10, 20, 30, 40, 49 are shown. Please report to [6, 7] for results comparisons.



**Fig. 6.** Results on *Carmap* sequence. First row, original sequence. Second row, results without rigidity constraint on the map. Third row, results with rigidity constraint on the map. Frames 1, 10, 11, 20, 30, 34 are shown. Please report to [7, 8] for results comparisons.

- [3] E. Mémin and P. Pérez, “Hierarchical estimation and segmentation of dense motion fields,” *IJCV*, vol. 46, no. 2, pp. 129–155, February 2002.
- [4] D. Cremers and S. Soatto, “Motion competition: A variational approach to piecewise parametric motion segmentation,” *IJCV*, vol. 62, no. 3, pp. 249–265, May 2005.
- [5] J.-M. Odobez and P. Bouthemy, “Direct incremental model-based image motion segmentation for video analysis,” *Signal Processing*, vol. 66, no. 2, pp. 143–155, 1998.
- [6] R. Dupont, N. Paragios, R. Keriven, and P. Fuchs, “Extraction of layers of similar motion through combinatorial techniques,” in *EMMCVPR*, November 2005.
- [7] J. Xiao and M. Shah, “Motion layer extraction in the presence of occlusion using graph cuts,” *PAMI*, vol. 27, no. 10, pp. 1644–1659, October 2005.
- [8] R. Dupont, O. Juan, and R. Keriven, “Robust segmentation of hidden layers in video sequences,” in *ICPR*, 2006.
- [9] Y. Boykov and M.-P. Jolly, “Interactive graph cuts for optimal boundary & region segmentation of objects in n-d images,” in *ICCV*, 2001.
- [10] Y. Boykov and V. Kolmogorov, “An experimental comparison of min-cut/max-flow algorithms for energy minimization in vision,” *PAMI*, vol. 26, no. 9, pp. 1124–1137, September 2004.
- [11] V. Kolmogorov and R. Zabih, “What energy functions can be minimized via graph cuts?,” *PAMI*, vol. 26, no. 2, pp. 147–159, February 2004.
- [12] Y. Boykov, O. Veksler, and R. Zabih, “Fast approximate energy minimization via graph cuts,” *PAMI*, vol. 23, no. 11, pp. 1222–1239, November 2001.