Occlusion-Based Accurate Silhouettes from Video Streams

Pedro M.Q. Aguiar, António R. Miranda, and Nuno de Castro

Institute for Systems and Robotics / Instituto Superior Técnico
Lisboa, Portugal
aguiar@isr.ist.utl.pt

Abstract. We address the problem of segmenting out moving objects from video. The majority of current approaches use only the image motion between two consecutive frames and fail to capture regions with low spatial gradient, *i.e.*, low textured regions. To overcome this limitation, we model explicitly: i) the *occlusion* of the background by the moving object and ii) the *rigidity* of the moving object across a set of frames. The segmentation of the moving object is accomplished by computing the Maximum Likelihood (ML) estimate of its silhouette from the set of video frames. To minimize the ML cost function, we developed a greedy algorithm that updates the object silhouette, converging in few iterations. Our experiments with synthetic and real videos illustrate the accuracy of our segmentation algorithm.

1 Introduction

Content-based representations for video enable efficient storage and transmission as well as powerful non-linear editing and manipulation [1]. The automatic segmentation of an image into regions that undergo different motions is a key step in the generation of content-based video representations. In this paper we address the problem of segmenting objects that exhibit a rigid motion across a set of frames.

A number of approaches to the segmentation of moving objects are found in the video coding literature. In fact, efficient video coding reduces temporal redundancy by predicting each frame from the previous one through motion compensation. Regions undergoing different movements are then compensated in different ways, according to their motion, see for example [2] for a review on very low bit rate video coding. The majority of these approaches are based on a single pair of consecutive frames and try to capture the moving object by detecting the regions that changed between the two co-registered images, see for example [3]. Since these methods were developed for image coding rather than for inferring high level representations, they often lead to inaccurate segmentation results. In particular, they fail to segment moving objects containing low textured regions because these regions are considered as unchanged, being then missclassified as belonging to the background.

A. Campilho and M. Kamel (Eds.): ICIAR 2006, LNCS 4141, pp. 816–826, 2006.

The more recent interest on the so-called layered representations for video [4,5,6,7,8] has motivated further work on motion-based segmentation. A number of approaches in the computer vision literature uses other cues besides motion, such as color and edges [9], or regularization priors [10]. In general, these methods lead to complex and time consuming algorithms.

Few approaches to motion segmentation use temporal integration, see [11,12,13,14] as examples. In [11,12], the images in the sequence are averaged after appropriate registration according to motion of the object. The silhouette of the moving object is estimated by detecting the regions of the current frame that are similar to the integrated image. This method overestimates the object silhouette unless the background is textures enough to blur completely the integrated image. The method in [13,14] exploits the occlusion of the background by the moving object—it estimates the silhouette of the object by integrating over time the intensity differences between the object and the background. This method succeeds even in low textured / low contrast scenes but it requires that the background is completely uncovered in the video clip.

We propose a new segmentation algorithm that exploits *occlusion* and *rigidity* without the drawback of the one in [13,14]. As in [13,14], we formulate the segmentation problem as the Maximum Likelihood (ML) estimation of the parameters involved in the video sequence model: the motions of the background, the motions of the object, the silhouette of the object, the intensity levels of the object (the object texture), and the intensity levels of the background (the background texture). The algorithm of [13,14] minimizes the ML cost function by computing, in two alternate steps, the estimates of: i) the object silhouette and ii) the background texture. We avoid the need to compute the background intensity levels at all pixels (and thus the requirement that the background is completely uncovered) by using the closed-form expression for the ML estimate of the background texture to derive the ML cost function as a function of the object silhouette alone. We develop a greedy algorithm that updates the silhouette converging in a small number of iterations.

Although our method is particularly tailored to the segmentation of rigid objects, it turns out also very useful to handle non-rigid ones. In fact, when processing videos showing non-rigid moving objects, the tracking procedures that cope with flexible silhouettes need an adequate initialization. Our method provides such an initialization because it will compute the *best rigid interpretation* of the scene, which suffices to segment out the moving objects. Finally, we remark that although our derivations assume scalar-valued images, *e.g.*, intensity of grey-level images, they are straightforwardly extended to vector-valued images, *e.g.*, multispectral images.

1.1 Paper Organization

In section 2 we formulate the segmentation problem as ML inference. Section 3 describes the ML cost function minimization procedure. In section 4 we outline how the algorithm is initialized. Section 5 contains experiments and section 6 concludes the paper.

2 Problem Formulation: Segmentation as Maximum Likelihood Inference

We consider 2D parallel motions, *i.e.*, all motions (translations and rotations) are parallel to the camera plane. We represent those motions by specifying time varying position vectors. These position vectors code rotation-translation pairs that take values in the group of rigid transformations of the plane, *i.e.*, the special Euclidean group SE(2). The vector $\mathbf{p}_f$ represents the position of the background relative to the camera in frame f. The vector $\mathbf{q}_f$ represents the position of the moving object relative to the camera in frame f. The image obtained by applying the rigid motion coded by the vector $\mathbf{p}$ to the image $\mathbf{I}$ is denoted by $\mathcal{M}(\mathbf{p})\mathbf{I}$, *i.e.*, pixel (x, y) of the image $\mathcal{M}(\mathbf{p})\mathbf{I}$ is given by $\mathcal{M}(\mathbf{p})\mathbf{I}(x, y) = \mathbf{I}(f_x(\mathbf{p}; x, y), f_y(\mathbf{p}; x, y))$, where $f_x(\mathbf{p}; x, y)$ and $f_y(\mathbf{p}; x, y)$ represent the coordinate transformation imposed by the 2D rigid motion coded by $\mathbf{p}$. We denote the inverse of $\mathcal{M}(\mathbf{p})$ by $\mathcal{M}(\mathbf{p}^{\#})$ and the composition of $\mathcal{M}(\mathbf{a})$ with $\mathcal{M}(\mathbf{b})$ by $\mathcal{M}(\mathbf{ab})$, *i.e.*, we have $\mathcal{M}(\mathbf{pp}^{\#})\mathbf{I} = \mathbf{I}$. For more details, see [13,14].

2.1 Observation Model

We consider a scene with a moving object in front of a moving camera. The pixel (x, y) of the image $\mathbf{I}_f$ belongs either to the background $\mathbf{B}$ or to the object $\mathbf{O}$. The image $\mathbf{I}_f$ is then modelled as

$$\mathbf{I}_f = \left\{ \mathcal{M}(\mathbf{p}_f^{\#})\mathbf{B} \left[\mathbf{1} - \mathcal{M}(\mathbf{q}_f^{\#})\mathbf{T} \right] + \mathcal{M}(\mathbf{q}_f^{\#})\mathbf{O}\, \mathcal{M}(\mathbf{q}_f^{\#})\mathbf{T} + \mathbf{W}_f \right\} \mathbf{H}, \quad (1)$$

where we make $\mathbf{I}_f(x, y) = 0$ for (x, y) outside the region observed by the camera. This is taken care of in (1) by the binary mask $\mathbf{H}$ whose (x, y) entry is such that $\mathbf{H}(x, y) = 1$ if pixel (x, y) is in the observed image $\mathbf{I}_f$ or $\mathbf{H}(x, y) = 0$ if otherwise. Naturally, $\mathbf{H}$ does not depend on the frame index f, since the motion of the camera is captured as background motion. In (1), $\mathbf{T}$ is the moving object silhouette—$\mathbf{T}(x, y) = 1$ if the pixel (x, y) belongs to the moving object or $\mathbf{T}(x, y) = 0$ if otherwise—and $\mathbf{W}_f$ stands for the observation noise, assumed Gaussian, zero mean, and white.

2.2 Maximum Likelihood Inference

Given a set of F video frames $\{\mathbf{I}_f, 1 \leq f \leq F\}$, we want to estimate the background texture $\mathbf{B}$, the object texture $\mathbf{O}$, the object silhouette $\mathbf{T}$, the camera poses $\{\mathbf{p}_f, 1 \leq f \leq F\}$, and the object positions $\{\mathbf{q}_f, 1 \leq f \leq F\}$. Using the observation model in (1) and the Gaussian white noise assumption, ML estimation leads to the minimization over all parameters of the functional

$$\begin{aligned} C\left(\mathbf{B}, \mathbf{O}, \mathbf{T}\left\{\mathbf{p}_f\right\}, \left\{\mathbf{q}_f\right\}\right) = \iint \sum_{f=1}^{F} \Big\{ & \mathbf{I}_f(x, y) \\ & - \mathcal{M}(\mathbf{p}_f^{\#})\mathbf{B}(x, y) \left[1 - \mathcal{M}(\mathbf{q}_f^{\#})\mathbf{T}(x, y) \right] \\ & - \mathcal{M}(\mathbf{q}_f^{\#})\mathbf{O}(x, y)\, \mathcal{M}(\mathbf{q}_f^{\#})\mathbf{T}(x, y) \Big\}^2 \mathbf{H}(x, y)\, dx\, dy, \end{aligned} \quad (2)$$

where the inner sum is over the full set of F frames and the outer integral is over all pixels. For details, see [13,14].

3 Maximum Likelihood Estimation: Greedy Algorithm

The minimization of the functional C in (2) with respect to (wrt) the set of constructs $\{\mathbf{B}, \mathbf{O}, \mathbf{T}\}$ and to the motions $\{\{\mathbf{p}_f\}, \{\mathbf{q}_f\}, 1 \leq f \leq F\}$ is a highly complex task. To obtain a computationally feasible algorithm, we decouple the estimation of the motion vectors from the determination of the constructs $\{\mathbf{B}, \mathbf{O}, \mathbf{T}\}$. This is reasonable from a practical point of view and is well supported by experimental results with real videos. We perform the estimation of the motions on a frame by frame basis by using known motion estimation methods [15]. After estimating the motions, we introduce the motion estimates into the ML cost C and minimize wrt the remaining parameters, *i.e.*, wrt the silhouette $\mathbf{T}$ of the moving object, the texture $\mathbf{O}$ of the moving object, and the texture $\mathbf{B}$ of the background.

We express the estimate $\widehat{\mathbf{O}}$ of the moving object texture and the estimate $\widehat{\mathbf{B}}$ of the background texture in terms of the object silhouette $\mathbf{T}$. By minimizing C in (2) wrt the intensity value $\mathbf{O}(x, y)$, we obtain the average of the pixels that correspond to the point (x, y) of the object. The estimate $\widehat{\mathbf{O}}$ of the moving object texture is then

$$\widehat{\mathbf{O}} = \mathbf{T}\, \frac{1}{F} \sum_{f=1}^{F} \mathcal{M}(\mathbf{q}_f) \mathbf{I}_f. \tag{3}$$

Minimizing the ML cost (2) wrt the intensity value $\mathbf{B}(x, y)$, we get the estimate $\widehat{\mathbf{B}}(x, y)$ as the average of the observed pixels that correspond to the pixel (x, y):

$$\widehat{\mathbf{B}} = \frac{\sum_{f=1}^{F} \left[\mathbf{1} - \mathcal{M}(\mathbf{p}_f \mathbf{q}_f^{\#}) \mathbf{T}\right]\, \mathcal{M}(\mathbf{p}_f) \mathbf{I}_f}{\sum_{i=f}^{F} \left[\mathbf{1} - \mathcal{M}(\mathbf{p}_f \mathbf{q}_f^{\#}) \mathbf{T}\right]\, \mathcal{M}(\mathbf{p}_f) \mathbf{H}}. \tag{4}$$

The estimate $\widehat{\mathbf{B}}$ of the background texture in (4) is the average of the observations $\mathbf{I}_f$ registered according to the background motion $\mathbf{p}_i$, in the regions $\{(x, y)\}$ not occluded by the moving object, *i.e.*, when $\mathcal{M}(\mathbf{p}_f \mathbf{q}_f^{\#}) \mathbf{T}(x, y) = 0$. The term $\mathcal{M}(\mathbf{p}_f) \mathbf{H}$ provides the correct averaging normalization in the denominator by accounting only for the pixels seen in the corresponding image.

We now replace the estimates $\widehat{\mathbf{O}}$ and $\widehat{\mathbf{B}}$, given by expressions (3,4), in the cost function (2), obtaining an expression for the ML cost function C in terms of a single unknown—the moving object silhouette $\mathbf{T}$, $C(\mathbf{T})$. This is an huge difference from the approach in [13,14], where only the estimate $\widehat{\mathbf{O}}$ is replaced in (2), leading to an expression for the ML cost function C in terms of $\mathbf{B}$ and $\mathbf{T}$, *i.e.*, $C(\mathbf{B}, \mathbf{T})$. In [13,14], the ML cost is minimized by using a two-step iterative algorithm that computes, in alternate steps, the minimum of $C(\mathbf{B}, \mathbf{T})$ wrt $\mathbf{B}$ for fixed $\mathbf{T}$, and the minimum of $C(\mathbf{B}, \mathbf{T})$ wrt $\mathbf{T}$ for fixed $\mathbf{B}$. This last step requires that (the previous estimate of) the background texture $\mathbf{B}$ is known at

all pixels, in particular it imposes that all background pixels occluded by the moving object are observed at least in one frame of the video sequence. Thus, the method of [13,14] does not deal with videos where the moving object only partially un-occludes the background, *i.e.* where some region of the background is occluded at all frames. In contrast, we propose to replace the expression of the background estimate $\widehat{\mathbf{B}}$ in terms of the object silhouette $\mathbf{T}$ into the ML cost C in (2), leading to an expression for $C(\mathbf{T})$ that is suitable to minimize wrt to the moving object silhouette $\mathbf{T}$ alone.

Replacing the estimates of $\mathbf{O}$ and $\mathbf{B}$, given by expressions (3) and (4), into the ML cost function (2), we get, after simple manipulations:

$$
\begin{aligned}
C\left(\mathbf{T}\right) = \iint \sum_{f=1}^{F} \Bigg\{ & \mathbf{I}_f(x,y) \\
& - \frac{\sum_{i=1}^{F} \left[\mathbf{1} - \mathcal{M}(\mathbf{p}_f^{\#}\mathbf{p}_i\mathbf{q}_i^{\#})\mathbf{T}\right] \; \mathcal{M}(\mathbf{p}_f^{\#}\mathbf{p}_i)\mathbf{I}_i}{\sum_{i=f}^{F} \left[\mathbf{1} - \mathcal{M}(\mathbf{p}_f^{\#}\mathbf{p}_i\mathbf{q}_i^{\#})\mathbf{T}\right] \; \mathcal{M}(\mathbf{p}_f^{\#}\mathbf{p}_i)\mathbf{H}} \left[1 - \mathcal{M}(\mathbf{q}_f^{\#})\mathbf{T}(x,y)\right] \\
& - \mathcal{M}(\mathbf{q}_f^{\#})\mathbf{T} \, \frac{1}{F} \sum_{i=1}^{F} \mathcal{M}(\mathbf{q}_f^{\#}\mathbf{q}_i)\mathbf{I}_i \Bigg\}^2 \mathbf{H}(x,y) \, dx \, dy \, . \qquad (5)
\end{aligned}
$$

We minimize this resulting cost $C(\mathbf{T})$ wrt its only argument $\mathbf{T}$ by using a greedy approach, in the spirit of several schemes that were successfully used to segment single images according to attributes like intensity, color, or texture, see the original energy minimization formulation of [16] and approaches that use variational methods [17], levels sets [18], partial differential equations [19], snakes [20,21], or active contours [22]. In our approach, given a previous estimate $\widehat{\mathbf{T}}_n$ of the moving object silhouette, the algorithm updates the estimate by including in $\widehat{\mathbf{T}}_{n+1}$ the neighboring pixels of $\widehat{\mathbf{T}}_n$ that lead to a decrease of the cost C and excluding the neighboring pixels that lead to an increase of C.

4 Initialization: Motion Detection

To initialize the segmentation algorithm, we need an initial guess of the silhouette of the object. Our experience has shown that the algorithm converges to the correct solution even when the initial guess of the silhouette is very far from the optimal estimate, for example when the initial guess is a single pixel in the interior of the object. However, the impact of a computationally simple initialization algorithm is high because, as it always happens with iterative algorithms, the closer is the initial guess to the correct solution, the faster is the convergence.

We compute the initial guess by using motion detection. To improve over simply detecting the motion between two frames, we merge silhouettes computed from several pairs of frames. The following example illustrates the procedure.

4.1 Synthetic Sequence 1

We synthesize a video sequence using an object texture that contains regions of almost constant intensity, *i.e.*, regions with low texture. Fig. 1 represents three frames of that synthetic video that shows a static background and a moving car. Note that, due to the low textured regions of the car, motion segmentation is not trivial for this video sequence, as referred in section 1.

Fig. 1. Synthetic video sequence 1

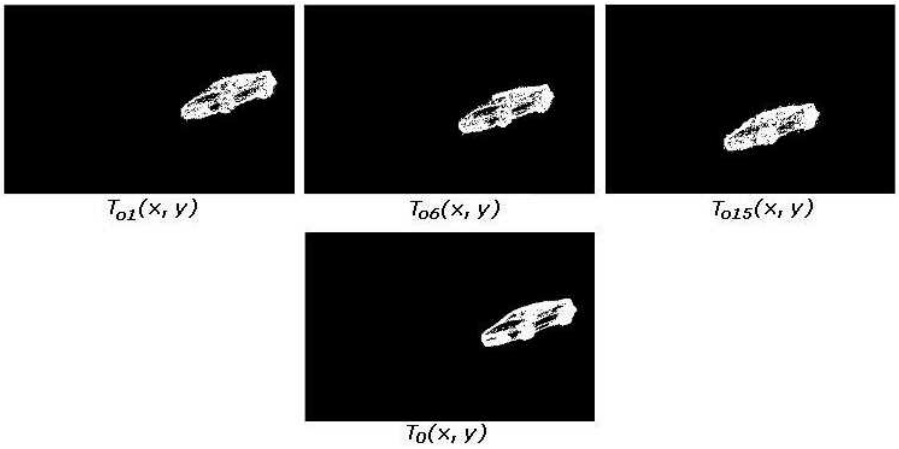

Fig. 2. Initial estimate of the silhouette of the moving car in the video in Fig. 1

In Fig. 2 we illustrate the initialization procedure for the first 20 frames of the video of Fig. 1. The top row contains pairwise estimates of the silhouette. These estimates are very incomplete due to the low texture of the car. The bottom image represents the initial guess of the silhouette obtained by merging the pairwise estimates. We see that this initial guess is more accurate than the pairwise estimates but it still misses a considerable number of pixels. Note that "filling-in" the regions that are missing in this initial guess by using spatial rules, *e.g.*, with morphological operations, is not trivial and requires the manual adaptation of several parameters in general dependent of the video sequence being processed.

5 Experiments

We report the results of our algorithm when segmenting two synthetic image sequences and one real video sequence.

In Fig. 3 we represent (left to right, top to bottom) the evolution of the estimate of the moving object silhouette, superimposed with its texture, for the synthetic video of Fig. 1. We see that, even for this low textured object, the estimate converges to the correct silhouette of the car. To better illustrate the behavior of the algorithm, we represent in Fig. 4, from left to right, the evolution of the estimate of the background texture. The left image of Fig. 4 shows the estimate at an early stage of the iterative process, *i.e.*, it shows an estimate that is blurred due to the still inaccurate estimate of the object silhouette. The right image of Fig. 4 demonstrates how the final estimate of the background texture is correct, *i.e.*, it is not blurred by the object texture. Note that, since in this

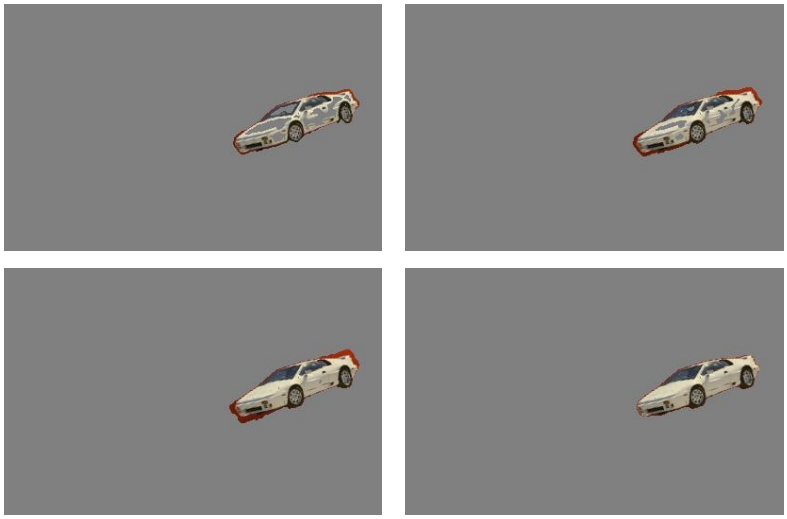

Fig. 3. Evolution of the estimate of the silhouette of the moving car for the video sequence in Fig. 1

Fig. 4. Evolution of the estimate of the background texture for the video sequence in Fig. 1

Fig. 5. Synthetic video sequence 2. Note that, since parts of the background are not seen at any frame (they are occluded by the moving object at all frames), the method of [13,14] can not be used to segment this video sequence.

Fig. 6. Background estimates for the video sequence in Fig. 5. The parts of the background that are not seen in any frame of the video sequence, are represented in black.

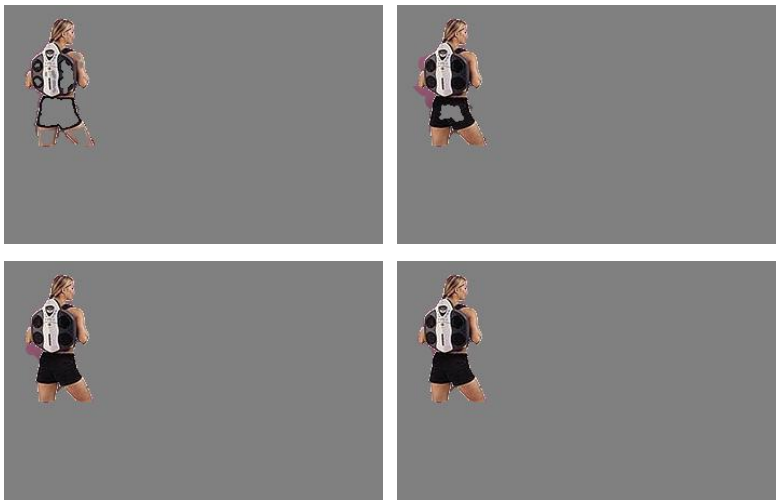

Fig. 7. Evolution of the estimate of the moving object silhouette for the video sequence in Fig. 5. In spite of the incomplete observation of the background, our method succeeded in segmenting accurately the moving object.

video clip the background was completely uncovered, the final estimate in the right image in Fig. 4 can be completely computed, *i.e.*, computed at all pixels.

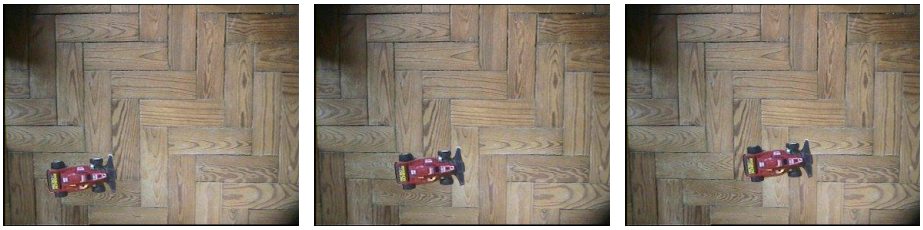

Fig. 8. Real video sequence

Fig. 9. Evolution of the estimate of the background texture from the video in Fig 8

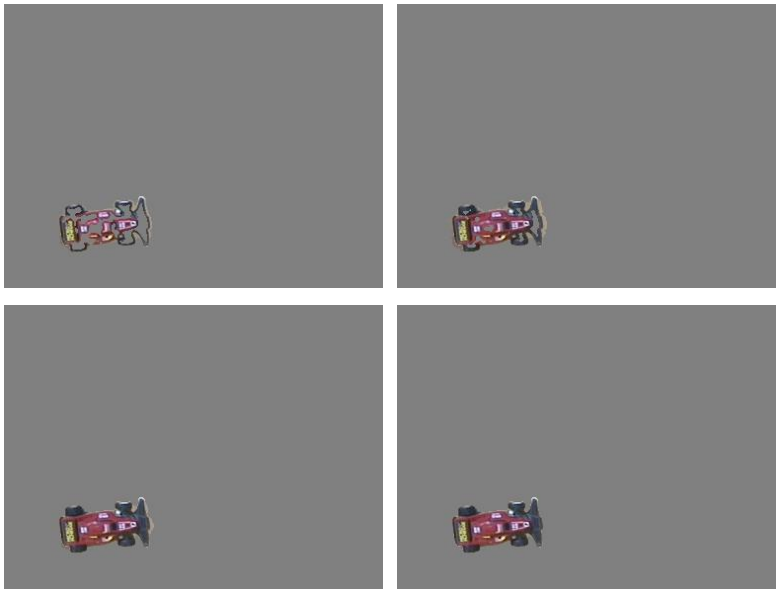

Fig. 10. Evolution of the estimate of the silhouette of the moving car from the video in Fig 8

5.1 Synthetic Sequence 2

We now synthesize a video sequence that shows a moving object with a more challenging shape. It also exhibits low textured regions. Fig. 5 shows three frames of this sequence. In this video the synthetic motion of the object is such that the background is not completely uncovered. The algorithm proposed in [13,14] to minimize the ML cost would then fail to segment this moving object. This is because the algorithm of [13,14] requires building a complete estimate of the background at intermediate steps, see discussion in section 1.

In Figs. 6 and 7 we represent the evolution of the estimates of the background texture and the moving object silhouette, respectively. Note that the background texture in the right image of Fig. 6 is not complete—we represent in black the pixels that, due to the occlusion by the moving object, were not observed in the video clip. As expected, our method is not affected by this covered background areas—we see from the bottom right image of Fig. 7 that our algorithm succeeded in accurately segmenting out the moving object in this video clip.

5.2 Real Video Sequence

We use a real video sequence that shows a moving car. Fig. 8 shows three frames from this video clip. Figs. 9 and 10 represent the evolution of the algorithm, demonstrating its good performance. See the evolution of the estimates of the background texture, in Fig. 9, and of the moving object silhouette, in Fig. 10.

6 Conclusion

We proposed a new algorithm to segment moving objects in video sequences. The algorithm exploits the *rigidity* of the object silhouette and the *occlusion* of the background by the moving object. Our experimental results illustrate the behavior of the algorithm and demonstrate its effectiveness.

References

1. Aguiar, P., Jasinschi, R., Moura, J., Pluempitiwiriyawej, C.: Content-based image sequence representation. In Reed, T., ed.: Digital Video Processing. CRC Press (2004) 7–72 Chapter 2.
2. Li, H., Lundmark, A., Forchheimer, R.: Image sequence coding at very low bitrates: A review. IEEE Trans. on Image Processing **3**(5) (1994)
3. Diehl, N.: Object-oriented motion estimation and segmentation in image sequences. Signal Processing: Image Communication **3**(1) (1991)
4. Jasinschi, R., Moura, J.: Content-based video sequence representation. In: Proc of IEEE Int. Conf. on Image Processing, Washigton D.C., USA (1995)
5. Sawhney, H., Ayer, S.: Compact representations of videos through dominant and multiple motion estimation. IEEE Trans. on Pattern Analysis and Machine Intelligence **18**(8) (1996)

6. Jasinschi, R., Moura, J.: Generative Video: Very Low Bit Rate Video Compression. U.S. Patent and Trademark Office, S.N. 5,854,856 (1998)
7. Tao, H., Sawhney, H., Kumar, R.: Dynamic layer representation with applications to tracking. In: Prof. of IEEE Int. Conf. on Computer Vision and Pattern Recognition, Hilton Head Island, South Carolina (2000)
8. Jojic, N., Frey, B.: Learning flexible sprites in video layers. In: Proc. of IEEE Int. Conf. on Computer Vision and Pattern Recognition, Hawaii (2001)
9. Dubuisson, M.P., Jain, A.: Contour extraction of moving objects in complex outdoor scenes. Int. Jounal of Computer Vision **14**(1) (1995)
10. Bouthemy, P., François, E.: Motion segmentation and qualitative dynamic scene analysis from an image sequence. Int. Jounal of Computer Vision **10**(2) (1993)
11. Irani, M., Peleg, S.: Motion analysis for image enhancement: Resolution, occlusion, and transparency. Journal of Visual Communications and Image Representation **4**(4) (1993) 324–335
12. Irani, M., Rousso, B., Peleg, S.: Computing occluding and transparent motions. Int. Journal of Computer Vision **12**(1) (1994)
13. Aguiar, P., Moura, J.: Maximum likelihood estimation of the template of a rigid moving object. In: Energy Minimization Methods in Computer Vision and Pattern Recognition. Springer–Verlag, LNCS 2134 (2001)
14. Aguiar, P., Moura, J.: Figure–ground segmentation from occlusion. IEEE Trans. on Image Processing **14**(8) (2005)
15. Bergen, J., et al.: Hierarchical model-based motion estimation. In: Proc of European Conf. on Computer Vision, Santa Margherita Ligure, Italy (1992)
16. Mumford, D., Shah, J.: Boundary detection by minimizing functionals. In: Prof. of IEEE Int. Conf. on Computer Vision and Pattern Recognition, San Francisco, CA, USA (1985)
17. Morel, J., Solimini, S.: Variational Methods in Image Segmentation. Birkhäuser, Boston (1995)
18. Malladi, R., Sethian, J., Vemuri, B.: Shape modeling with front propagation: A level set approach. IEEE Trans. on Pattern Analysis and Machine Intelligence **17**(2) (1995) 158–175
19. Sapiro, G.: Geometric Partial Differential Equations and Image Analysis. Cambridge University Press (2001)
20. Kass, M., Witkin, A., Terzopoulos, D.: Snakes: Active contour models. Int. Journal of Computer Vision **1**(4) (1988) 321–331
21. Caselles, V., Kimmel, R., Sapiro, G.: Geodesic snakes. Int. Journal of Computer Vision **22** (1997) 61–79
22. Chan, T., Vese, L.: Active contours without edges. IEEE Trans. on Image Processing **10**(2) (2001) 266–277