Robust Segmentation of Hidden Layers in Video Sequences

Romain Dupont Olivier Juan Renaud Keriven

CERTIS Laboratory,
Ecole des Ponts, Paris, France
{*dupont, juan, keriven*}*@certis.enpc.fr*

Abstract

In this paper, we propose a novel and robust method for extracting motion layers in video sequences. Taking advantage of temporal continuity, our framework considers both the visible and the hidden parts of each layer in order to increase robustness. Moreover, the hidden parts of the layers are recovered*, which could be of great help in many high level vision tasks. Modeling the problem as a labeling task, we state it in a MRF-optimization framework and solve it with a graph-cut algorithm. Both synthetic and real video sequences show a visible layers extraction comparable to the one usually performed by state of the art methods, as well as a novel and successful segmentation of hidden layers.*

1. Introduction

We consider the extraction of the layers composing a video sequence, each of them being approximated by a planar set of objects having the same parametric motion. This well studied representation (see [1, 2, 3, 4, 7, 9, 11, 12, 13, 14, 15]) offers a good trade-off between low- and high-level of information for numerous applications, such as robust motion segmentation, efficient video compression, 3D reconstruction of urban scenes, etc. The main issues addressed in this context are the estimation of the motion of the layers, the outliers and occlusion detection, the determination of the number of layers, the choice of regularization criteria and the accuracy and robustness of the segmentation.

In [16], Xiao and Shah present a method based on temporal constraints between a frame and its successors ($1 \mapsto 2$, $1 \mapsto 3$, $1 \mapsto 4$, ...) that takes into account what they call occlusions (actually, point modeling two distinct phenomena: (i) objects becoming hidden and (ii) noisy point with impossible tracking). Their method does not intrinsically give smooth segmentations from one frame to the other as frames are processed independently.

On the contrary, our method takes advantage of temporal information for the whole sequence. Indeed, it simultaneously processes all the sequence considering temporal constraints between successive frames $1 \mapsto 2 \mapsto 3 \mapsto 4 \mapsto ...$, guaranteeing a smooth labeling. Furthermore, it explicitly recovers the hidden parts of the layers, that can disappear behind an another one and re-appear a few frames later: *a disappearing point is not only detected like in [16] but also tracked while being hidden until it re-appears*! Finally, tracking both visible and hidden parts of layers reduces segmentation ambiguities, namely the number of *undefined* points (see further).

Hidden layers. For each pixel, we consider its corresponding visible layer and all hidden layers if any. Given n, the number of layers, we associate each pixel $\mathbf{x}$ with its label $l_{\mathbf{x}} = (v_{\mathbf{x}}, \mathbf{h}_{\mathbf{x}}) \in \mathcal{L}$, with $\mathcal{L} = (\mathcal{V} \times \mathcal{H}) \setminus \mathcal{F}$, where $\mathcal{V} = [1, n] \cup \{\varnothing_{\mathcal{V}}\}$ is the visible space, $\mathcal{H} = \{\mathbf{false}, \mathbf{true}\}^n$ is the hidden one and $\mathcal{F}$ refers to forbidden combinations (see further). The special label $\varnothing_{\mathcal{V}}$ corresponds to an indetermination on the visible layer choice (*undefined pixels* or *"outliers"*). The i^{th} coordinate $\mathbf{h}^i_{\mathbf{x}}$ of vector $\mathbf{h}_{\mathbf{x}}$ indicates the hidden state of the i^{th} layer ($\mathbf{true}$ if hidden, $\mathbf{false}$ if visible or non present). For a given pixel, a layer cannot be both visible and hidden, i.e. $\mathbf{h}^{v_{\mathbf{x}}}_{\mathbf{x}} \neq \mathbf{true}$: $\mathcal{F}$ is the set of such forbidden cases. Figure 1 illustrates such a labeling.

Motion model. $\mathcal{T}^t_v$ will denote the estimated motion of layer v between frames t and $t+1$. No motion is associated to layer $\varnothing_{\mathcal{V}}$. Our experiments use classical projective motions, thus approximates the scene by three-dimensional plane objects, although any other model could be used (e.g. affine). Motion estimation follows our previous work [7] and will not be detailed here, though any other equivalent method could be used.

Initialization. Our method is initialized with n pre-

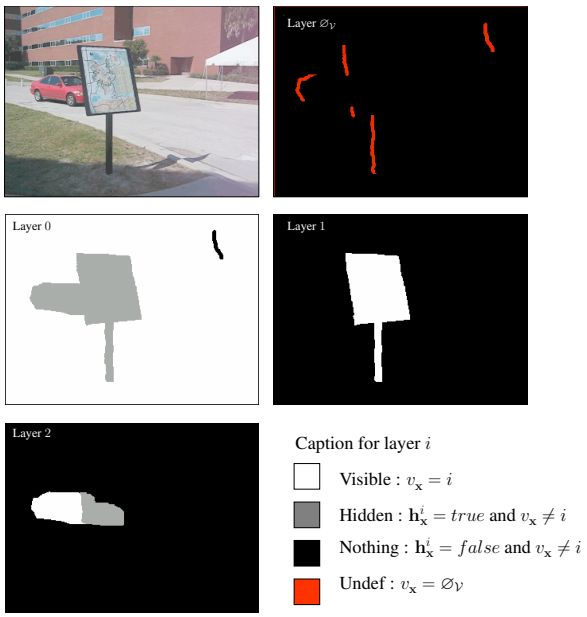

Figure 1. Example of labeling. Note that these images **are not the results** obtained by our algorithm but an example of what could be a reasonable segmentation.

computed layers (accurate or not), obtained through pre-existent methods like the ones in [7, 16]. When the correspondences between the layers of successive frames is not explicitly given by this initial segmentation, we recover it easily, associating a layer v at time t to the one at time $t+1$ that most overlaps its image through $\mathcal{T}_v^t$.

Overall process. Our method consists in alternating, until some stabilization: (i) layer segmentation and (ii) refinement of the motion parameters from the visible part of the layers (which, again, will not be detailed here).

The reminder of this paper is organized in the following way. Section 2 presents the energy used for classification. Section 3 provides some important information about the implementation and shows results on both synthetic and real data. The last section gives some conclusion and future directions.

2 Classification

Given T frames, n layers, and $\mathcal{T}_v^t (v \in [1,n], t \in [1,T])$ their motion models [1], we consider the labeling problem consisting in determining a function $L : (\mathbf{x}, t) \mapsto l_{\mathbf{x}}^t = (v_{\mathbf{x}}^t, \mathbf{h}_{\mathbf{x}}^t) \in \mathcal{L}$. We plug the problem into a variational framework and will design in the sequel an energy that L should minimize. Note that we consider a constant number of layers throughout the sequence. Such a limitation could be relaxed through appropriate methods.

[1] when explicitly needed, the frame number t will be indicated by a superscript

2.1 Motion energy

The motion energy is based on visible parts of the layers and is indeed related to the images ("*data term*"). The *forward* motion residual $r_v(\mathbf{x})$ for the pixel $\mathbf{x}$ under motion $\mathcal{T}_v$ is defined by:

$$r_v^t(\mathbf{x}) = \left\| I^t(\mathbf{x}) - I^{t+1}(\mathcal{T}_v^t(\mathbf{x})) \right\| \tag{1}$$

where I^t is the image at time t. To reduce the influence of high motion residuals, we apply a smoothed Heaviside operator ψ given by:

$$\psi(r_v) = \tan^{-1}\left(r_v^2 - \tau\right) + \pi/2 \tag{2}$$

We define a labeling cost function d_I by:

$$d_I(l_{\mathbf{x}}, \mathbf{x}) = \begin{cases} \psi\left(r_{v_{\mathbf{x}}}(\mathbf{x})\right) & \text{if } v_{\mathbf{x}} \in [1,n] \\ \psi_{undef} & \text{if } v_{\mathbf{x}} = \varnothing_V \end{cases} \tag{3}$$

where the parameter ψ_{undef} adjusts the classification of pixels as undefined. The *forward* motion energy E_{FM}^t is then, for a given frame t:

$$E_{FM}^t(L) = \int_\Omega d_I(l_{\mathbf{x}}^t, \mathbf{x}) d\mathbf{x} \tag{4}$$

where Ω is the image domain. To increase robustness, we also consider the *backward* motion residual (as in [12]) and its associated energy noted $E_{BM}^t(L)$. It is defined similarly, considering frame $t-1$ instead of frame $t+1$ and the reverse motion $(\mathcal{T}_v^t)^{-1}$ instead of $\mathcal{T}_v^t$.

2.2 Spatial regularization

As in every noisy and under-constrained problem, spatial regularization has to be introduced. Both visible and hidden parts of the layers are regularized through the following energy:

$$E_S^t(L) = \iint_{\Omega^2} \phi(||\mathbf{x} - \mathbf{y}||) d_S^t\left(l_{\mathbf{x}}^t, l_{\mathbf{y}}^t\right) d\mathbf{y} d\mathbf{x} \tag{5}$$

where ϕ is some kernel (e.g Gaussian) and $d_S^t(.,.)$ is a dissimilarity measure between two labels. Discontinuous labels for both visible and hidden layers must be penalized. We encourage also the frontier of the layer to belong to pixels with high image gradient. This gives the following function:

$$d_S^t(l_{\mathbf{x}}, l_{\mathbf{y}}) = \mu_V \mathcal{I}\left(v_{\mathbf{x}} \neq v_{\mathbf{y}}\right) \exp\left(-\frac{\|I^t(\mathbf{x}) - I^t(\mathbf{y})\|^2}{2\sigma^2}\right) + \mu_H \sum_{i=1}^{n} \mathcal{I}\left(\mathbf{h}_{\mathbf{x}}^i \neq \mathbf{h}_{\mathbf{y}}^i\right) \tag{6}$$

where $\mathcal{I}(i)$ equals 1 if i is true, 0 otherwise, σ is the standard deviation of the norm of the gradient of the images, and (μ_V, μ_H) some constants adjusting spatial regularization with respect to the other energy terms.

2.3 Temporal constraints

Temporal constraints are designed for both temporal smoothness and temporal consistency between visible and hidden layers. To this end, using motion information, we penalize discontinuous labeling between frames. To simplify notations, we note $\mathbf{x}_i = \mathcal{T}_i^t(\mathbf{x})$ the image of $\mathbf{x}$ in frame $t+1$ through the motion of layer i at time t. Our *forward* temporal energy is written as follows:

$$E_{FT}^t(L) = \int_\Omega \Big[\mathcal{I}(v_\mathbf{x} \neq \varnothing_\mathcal{V}) d_V\left(l_\mathbf{x}^t, l_{\mathbf{x}_{v_\mathbf{x}}}^{t+1}\right) + \sum_{i=1}^n \mathcal{I}(\mathbf{h}_\mathbf{x}^i = \mathbf{true}) d_H^i\left(l_\mathbf{x}^t, l_{\mathbf{x}_i}^{t+1}\right)\Big] d\mathbf{x} \quad (7)$$

where $d_V(.,.)$ and $d_H^i(.,.)$ are dissimilarity measures given by:

$$d_V(l_\mathbf{x}, l_\mathbf{y}) = \begin{cases} 0 & \text{if } v_\mathbf{x} = v_\mathbf{y} \\ \lambda_H & \text{if } \mathbf{h}_\mathbf{y}^{v_\mathbf{x}} = \mathbf{true} \\ \lambda_D & \text{otherwise} \end{cases} \quad (8)$$

and:

$$d_H^i(l_\mathbf{x}, l_\mathbf{y}) = \begin{cases} 0 & \text{if } \mathbf{h}_\mathbf{y}^i = \mathbf{h}_\mathbf{x}^i \\ \lambda_V & \text{if } v_\mathbf{y} = i \\ \lambda_D & \text{otherwise} \end{cases} \quad (9)$$

where λ_H, λ_V and λ_D respectively penalize the following events: hiding, re-appearing, and completely disappearing. It can be shown (see [6]) that λ_D has to be chosen greater than λ_V and λ_H.

As in the data term, we also consider *backward* constraints, leading to a symmetric temporal energy E_{BT}^t.

2.4 Overall energy

Our overall energy to extract the optimal partition of the T images is finally:

$$E(L) = \sum_{t=1}^T [\underbrace{E_{FM}^t(L) + E_{BM}^t(L)}_{\text{data term (motion)}} + \underbrace{E_S^t(L)}_{\text{spatial regularization}} + \underbrace{\left(E_{FT}^t(L) + E_{BT}^t(L)\right)}_{\text{temporal constraints}}] \quad (10)$$

3 Energy minimization and Results

Implementation. We plug our spatially continuous energy minimization problem into a discrete Markov Random Field framework [8]. The global energy (EQ. 10) is discretized considering a 4- or 8- neighborhood for the spatial constraints. Due to its efficiency, we use the *alpha*-expansion [2] algorithm [5, 10].

Even then, labeling cannot be achieved in reasonable time using a straightforward *alpha*-expansion since the number of possible labels $(v, \mathbf{h})$ increases dramatically with the number of layers: $(n+2)2^{n-1}$ possible expansions! However the problem could be circumvented limiting *alpha*-expansions to a sub-space of $\mathcal{L}$, considering only a change of the visible layer and one hidden layer, i.e. $(v, \mathbf{h}^i)$-expansions for successive choices of i. Using this approach, we reduce the number of optimization steps to $2n^2$ (see [6] for details), yielding in practice to acceptable minimization times, without modifying noticeably the segmentation.

The corresponding graph is a three-dimensional one, the third dimension being time. The data and spatial regularization terms of the energy are standard in the graph-cut framework. During a $(v, \mathbf{h}^i)$-expansion, the *backward* and *forward* spatial constraints yield links between each pixel $\mathbf{x}$ at time t and 8 other pixels (see [6]): $\mathbf{x}_v$, $\mathbf{x}_{\mathbf{h}^i}$, $\mathbf{x}_{v_\mathbf{x}}$ and $\mathbf{x}_{\mathbf{h}_\mathbf{x}^i}$ at time $t+1$ and the 4 similar pixels at time $t-1$.

Synthetic sequence. Figure 2 shows the results obtained on a synthetic sequence ($n = 3$). Throughout the sequence, the proportion of misclassified visible pixels is 0.06% and the proportion of pixels where the complete label l (visible *and* hidden parts) is incorrect is also 0.06%: for each pixel, classification fails or succeeds globally. Note that in this particular sequence, no pixel is classified as undefined. Indeed, only noise or aliasing could generate such pixels. Because hidden parts are modelized, the undefined label do not account anymore for points that become hidden like in [16].

Real sequences. As a first step[3] toward comparing our results to state of the art methods like [7, 16], we show the results obtained for a real sequence (fig. 3).One can see that the segmentation of the visible layers is comparable to the usually obtained segmentation. Note that the wheels of the car are sometime classified as undefined because the number n of layers is fixed too small (the wheels have their own motion). A splitting/merging approach could be used to choose n dynamically. We are in the process of implementing this.

Moreover, our goal was to extract the hidden parts of the layers and this is correctly done. Continuous labeling between frames is obtained, providing non-disrupted segmentation throughout the sequences. Again, note that the number of undefined pixels is rather small: unlike in [16] where these pixels code also for points that are going to be

[2] One can easily check that d_S, d_V and d_H are sub-modular functions with $d(l,l) = 0$ (see [6]).

[3] No ground truth is provided here!

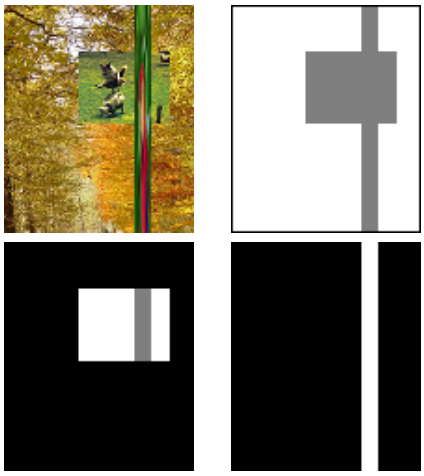

Figure 2. A synthetic sequence. From top to bottom, left to right: original sequence, layers 1, 2 and 3 (white=visible, grey=hidden) (Note: on this particular image of the sequence, no pixel is classified as undefined)

hidden, in our method $v_{\mathbf{x}} = \varnothing_V$ only stands for a lack of image information (e.g. too much noise).

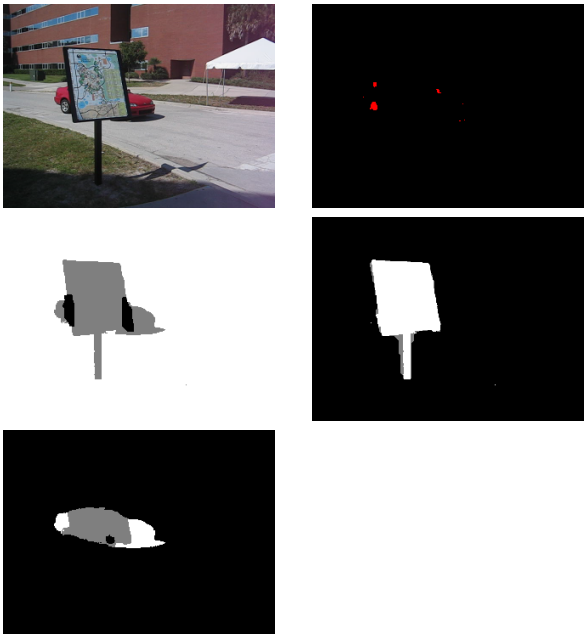

Figure 3. Carmap sequence. From top to bottom, left to right: original sequence, undefined pixels (in red), layers 1, 2 and 3 (white=visible, grey=hidden).

4 Conclusion and discussion

We have presented a novel global optimization process for motion layer segmentation in a video sequence. Considering the hidden parts of the layers, we achieve a continuous labeling, even is case of occlusion: when hidden, a point is tracked until reappearance. Ongoing work includes dealing with (i) processing longer sequences through shifting windows, (ii) more robustness thanks to multi-scale analysis in time and (iii) coping with a robust determination of a variable number of layers.

References

[1] S. Ayer and H. S. Sawhney. Layered representation of motion video using robust maximum-likelihood estimation of mixture models and mdl encoding. In *ICCV*, page 777, 1995.

[2] M. J. Black and P. Anandan. The robust estimation of multiple motions: parametric and piecewise-smooth flow fields. *Comput. Vis. Image Underst.*, 63(1):75–104, 1996.

[3] M. J. Black and A. D. Jepson. Estimating optical flow in segmented images using variable-order parametric models with local deformations. *IEEE Trans. Pattern Anal. Mach. Intell.*, 18(10):972–986, 1996.

[4] P. Bouthemy and E. Francois. Motion segmentation and qualitative dynamic scene analysis from an image sequence. *Int. J. Comput. Vision*, 10(2):157–182, 1993.

[5] Y. Boykov, O. Veksler, and R. Zabih. Fast approximate energy minimization via graph cuts. *IEEE Trans. Pattern Anal. Mach. Intell.*, 23(11):1222–1239, 2001.

[6] R. Dupont, O. Juan, and R. Keriven. Robust segmentation of hidden layers in video sequences. Technical Report 06-21, Certis / ENPC, 2006.

[7] R. Dupont, N. Paragios, R. Keriven, and P. Fuchs. Extraction of layers of similar motion through combinatorial techniques. In *EMMCVPR*, 2005.

[8] S. Geman and D. Geman. Stochastic Relaxation, Gibbs Distributions, and the Bayesian Restoration of Images. *IEEE. Trans. on PAMI*, 1984.

[9] Q. Ke and T. Kanade. A robust subspace approach to layer extraction. In *MOTION '02: Proceedings of the Workshop on Motion and Video Computing*, page 37, 2002.

[10] V. Kolmogorov and R. Zabih. What energy functions can be minimized via graph cuts? In *ECCV*, pages 65–81, 2002.

[11] M. P. Kumar, P. H. S. Torr, and A. Zisserman. Learning layered motion segmentations of video. In *ICCV*, 2005.

[12] J.-M. Odobez and P. Bouthemy. Direct incremental model-based image motion segmentation for video analysis. *Signal Processing*, 66(2):143–155, 1998.

[13] A. D. J. Shanon X. Ju, Michael J. Black. Skin and bones: Multi-layer, locally affine, optical flow and regularization with transparency. In *CVPR*, page 307, 1996.

[14] J. Wang and E. Adelson. Layered representation for motion analysis. In *CVPR93*, pages 361–366, 1993.

[15] Y. Weiss. Smoothness in layers: Motion segmentation using nonparametric mixture estimation. In *CVPR*, 1997.

[16] J. Xiao and M. Shah. Accurate motion layer segmentation and matting. In *CVPR*, pages 698–703, 2005.