Motion Segmentation Using a K-nearest-Neighbor-Based Fusion Procedure of Spatial and Temporal Label Cues

Pierre-Marc Jodoin and Max Mignotte

Université de Montréal
Département d'Informatique et de Recherche Opérationnelle (DIRO),
P.O. Box 6128, Studio Centre-Ville, Montréal, Québec, H3C 3J7.
E-mail: {jodoinp/mignotte}@iro.umontreal.ca

Abstract. Traditional motion segmentation techniques generally depend on a pre-estimated optical flow. Unfortunately, the lack of precision over edges of most popular motion estimation methods makes them unsuited to recover the exact shape of moving objects. In this contribution, we present an original motion segmentation technique using a K-nearest-neighbor-based fusion of spatial and temporal label cues. Our fusion model takes as input a spatial segmentation of a still image and an estimated version of the motion label field. It minimizes an energy function made of spatial and temporal label cues extracted from the two input fields. The algorithm proposed is intuitive, simple to implement and remains sufficiently general to be applied to other segmentation problems. Furthermore, the method doesn't depend on the estimation of any threshold or any weighting function between the spatial and temporal energy terms, as is sometimes required by energy-based segmentation models. Experiments on synthetic and real image sequences indicate that the proposed method is robust and accurate.

1 Introduction

Motion segmentation is one of the most studied research areas in computer vision. It refers to the general task of labeling image regions that contain uniform displacement. Consequently, motion segmentation has often be related to motion estimation. Actually, a common way to segment an image sequence is to estimate an optical flow field and then segment it into a set of regions with uniform displacement. Such an approach is sometimes called *motion-based* [1] since segmentation is performed on the basis of displacement vectors only. This kind of segmentation is rather easy to implement and generates more accurate results than say, an 8×8 block classification-segmentation procedure. However, motion-based approaches are known to depend on the accuracy of an optical flow field which isn't reliable over textureless and/or occluded areas. Consequently, motion-based algorithms are doomed to return imprecise results, especially around edges of moving objects.

To help motion segmentation converge toward more precise solutions (i.e., solutions in which the contour of segmented regions fit the silhouette of the moving objects), some include spatial constraints to the segmentation process. These constraints are often edges or regions extracted from one or more image frames. Motion segmentation approaches with spatial constraints are often called *spatio-temporal techniques*. These techniques are generally slower than motion-based approaches, but generate more precise segmentation results.

The approach we propose is based on a K-nearest-neighbor-based fusion procedure that mixes spatial and temporal data taken from two input label fields. The first one is a *spatial segmentation* which contains regions of uniform brightness while the second label field is an estimated version of the *motion label field* we will search to refine. The two segmentation maps are obtained with an unsupervised Markovian procedure. Our fusion method works with an iterative optimization algorithm called ICM (Iterative Conditional Mode) [2] whose mode (the maximum local energy for each site at each iteration) is obtained with a K-nearest neighbor algorithm. The result returned by

our fusion model is a label field that exhibits uniform regions in the sense of brightness and motion.

The rest of the paper is organized as follows. In Section 2, we present some motion segmentation techniques recently proposed by the computer vision community before section 3 describes the proposed technique. The Markovian method we use to generate the two input label fields is discussed in Section 4 while the overall algorithm we proposed is summarized in section 5. Section 6 presents results produced by our method while concluding remarks are presented in Section 7.

2 Previous Work

A great number of papers have been published in motion segmentation during the past two decades [1, 3]. Among the most popular *motion-based* approaches are the ones using parametric motion models [1]. The goal of these motion segmentation methods is to jointly estimate motion models and their associate motion regions. To this end, the motion regions and the motion model parameters are generally estimated in two steps [4] that are iterated until convergence. The first step consists in estimating the motion model parameters according to a pre-estimated optical flow field and the current motion label field [5, 6]. By opposition, the second step consists in estimating new motion regions while the motion models are kept unchanged. Tekalp [7, 8] summarizes these two steps with his *Maximum Likelihood* (ML) and *Maximum a Posteriori* (MAP) procedures. The difference between the former and the latter is the use of an *a priori* energy function that helps smoothing the resulting motion label field.

To our knowledge, Murray and Buxton [9] were the first to embed motion segmentation in a statistical framework using a Markov random field (MRF) model and a Bayesian criterion (a MAP criterion). Their technique uses quadratic motion models and represents the segmentation field with a Gibbs distribution whose energy is optimized with a Simulated Annealing (SA) algorithm. A few years later, Bouthemy and Francois [4] presented a motion-based segmentation approach relying on 2D affine models, used to detect moving objects in a scene observed by a moving camera. As for Murray and Buxton's method [9], they proposed a model based on a MAP criterion but include a temporal link between successive partitions to ensure temporal coherence. Bouthemy and Francois uses an ICM optimization to find the solution.

Other authors use motion segmentation to separate the scene into *moving layers* [10]. A well known iterative approach is the one proposed by Wang and Adelson [11]. The algorithm starts by estimating an optical flow field and subdivides the current frame into a predetermined number of square blocks. Affine motion models are then fitted over each block to get an initial set of motion models. Since the number of initial models is larger then the number of layers, the models are merged together with a K-means clustering method. Some layers can be split afterward to preserve spatial coherency.

Others have proposed segmentation models based on multiple features, such as brightness and motion. They are often refereed to as *spatio-temporal* segmentation techniques. In this context, Black [12] presented an incremental approach with constraint on intensity and motion while accounting for discontinuity. Its approach is based on a MRF and minimizes a three- term energy function using a stochastic relaxation technique. Altunbasak *et al.* [13] presented a motion segmentation approach working at a region level. As a first step, they independently compute a motion-based partition and a color-based partition. Assuming that color regions are more accurate than the motion regions, a region-based motion segmentation is performed, whereby all sites contained in a color region are assigned a single motion label. Bergen and Meyer [14] show how to use a still image segmentation combined with robust regres-

sion to eliminate error due to occlusion. This technique computes depth cues on the basis of motion estimation error.

Finally, a recent paper proposed by Khan and Shah [15] presents a MAP framework that softly blends color, position and motion cues to extract motion layers. Each cue has its own probability density function (PDF). These PDF are combined together with feature weights that give more or less importance to a cue depending on some defined observations.

3 Our Method

Our motion segmentation procedure takes as input two label fields. The first one is a spatial partition of a frame at time t (I^t) while the second one is an estimated version of the motion partition (cf. Fig.1). In our application, these two label fields –called respectively r and $x^{[0]}$– are estimated separately with an unsupervised Markovian procedure (although any other valid segmentation approaches would do the trick). The Markovian framework used in this paper is presented in Section 4.

Once r and $x^{[0]}$ have been computed, they are fed to a K-nearest-neighbor-based fusion procedure. This procedure –which is the core of our contribution– blends together spatial and temporal label cues to generate a partition with uniform regions in the sense of brightness *and* motion. In other words, this fusion procedure optimizes an energy function made of spatial and motion label terms extracted from the two input label fields. Details on this function and the optimization procedure are presented in Section 5.

Compared to previous methods, our approach has legitimate advantages. To start off with, our solution is unsupervised and, as opposed to [11] and [15], doesn't depend on any threshold or weighting function that might change from one sequence to another. Secondly, our method is stable and doesn't generates unexpected results when its parameters are tweaked. For example, as opposed to [13] that needs an accurate spatial partition, our method reacts well when r and/or $x^{[0]}$ lacks precision. Finally, our method is simple to implement and remains sufficiently general to be applied to other segmentation problems.

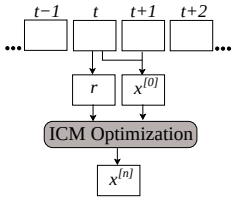

Fig. 1. *Schematic representation of our approach. From two frames at times $t-1$ and t, a spatial and a motion label field (r and $x^{[0]}$) are estimated. These label fields are then fed to the K-nearest neighbor fusion procedure (ICM optimization) that returns a partition ($x^{[n]}$) in which regions are uniform in the sense of brightness and motion.*

4 Markovian Segmentation

Given $Z = \{X, Y\}$, a pair of random fields where $X = \{x_s, s \in S\}$ and $Y = \{\boldsymbol{y}_s, s \in S\}$, represent respectively the label field and observation field defined on $S = \{s = (i, j)\}$, a 2D lattice of N sites. Here, Y (an image frame I^t or a vector field v) is known *a priori* whereas X has to be estimated. Each x_s takes a value in $\Gamma = \{1, \ldots, m\}$, where m corresponds to the number of classes of the segmentation map while $\boldsymbol{y}_s$ is a vector made of real elements.

Segmentation can be viewed as a statistical labeling problem, i.e., a problem where each observation vector $\boldsymbol{y}_s$ needs to be associated to the *best* class $x_s \in \Gamma$. Thus,

inferring a label field can be seen as an optimization problem that searches for *the best* x in the sense of a given statistical criterion. Among the available statistical criterion, the *Maximum a posteriori* states that a label field x is *optimal* according to y when it maximizes the *a posteriori* PDF $P(x|y)$. In this way, x is optimal whenever $x = \arg\max_x P(x|y)$ [2].

Because $P(x|y)$ is often complex and/or undefined, it is common to assume that X and Y are MRFs. In this way, this posterior distribution can be defined by a Gibbs distribution of the form $P(X|Y) \propto \exp -U(X,Y)$ where $U(X,Y)$ is an *energy* function [2]. From Bayes theorem [16], the a posteriori distribution can be represented as $P(X|Y) \propto \exp\{-(U_1(X,Y) + U_2(X))\}$ where U_1 and U_2 are the likelihood and prior energy functions.

By assuming independence between each random variable $\boldsymbol{Y}_s$ (i.e., $P(Y|X) = \prod_{s\in S} P(\boldsymbol{Y}_s|X_s)$), the corresponding posterior energy to be minimized is

$$U(X,Y) = \sum_{s\in S} \Bigg(\underbrace{\Psi_s(x_s, \boldsymbol{y}_s)}_{U_1(x_s,y_s)} + \underbrace{\sum_{<s,t>} \beta\big[1 - \delta_{x_s,x_t}\big]}_{U_2(x_s)} \Bigg), \tag{1}$$

where U_2 is an isotropic Potts model. Here, $\delta_{a,b}$ is the Kronecker function (returns 1 if $a = b$ and 0 elsewhere), β is a constant, $<s,t>$ is the set of binary *cliques* that includes s, and $\Psi_s(x_s, \boldsymbol{y}_s) = -\ln P(\boldsymbol{y}_s|x_s)$. Notice that the cliques are defined on a second-order neighborhood.

The conditional distribution $P(\boldsymbol{y}_s|x_s)$ models the distribution of the observed data $\boldsymbol{y}_s$ given a class x_s. In this paper, this distribution is modeled with a Normal law which depends on the two parameters $(\boldsymbol{\mu}_{x_s}, \Sigma_{x_s})$. Since there are m different classes, there are m different Normal laws and a total of $2m$ parameters $\Phi = [(\mu_1, \sigma_1) , \ldots , (\mu_m, \sigma_m)]$. Because these parameters are initially unknown, they have to be estimated. To this end, we resort to an iterative method called Iterated Conditional Estimation (ICE) [17].

Markovian Spatial Segmentation The spatial label field r is obtained by segmenting image frame I^t with a Markovian procedure based on the the framework presented in the previous Section. Here, I^t stands for the observation field y while $\boldsymbol{y}_s$ is a singleton that takes its value in $\{0, \ldots, 255\}$. For RGB color images, the brightness of each site is obtained by simply computing the average value of the three channels, i.e. $y_s = (I^t_{s_r} + I^t_{s_g} + I^t_{s_b})/3$.

Markovian Motion Segmentation The second label field fed to the optimization procedure is a motion-based partition called $x^{[0]}$. Although this partition could be obtained with any method presented in Section 2, we decided to use an unsupervised statistical Markovian procedure. Here, an optical flow field v computed with an iterative version [18] of Simoncelli *et al.*'s algorithm [19] stands for the observation field y. Every element $\boldsymbol{y}_s$ is thus a two-dimensional real vector. For every sequence we have tested, v was computed with a two-level pyramid and an integration window of size 7×7 [18].

5 *K*-Nearest-Neighbor-Based Fusion

Once r and $x^{[0]}$ have been estimated, they are fed to the K-nearest-neighbor-based fusion approach as shown in Fig.1. This procedure seeks a motion label field x made of regions uniform in the sense of brightness (r) and motion ($x^{[0]}$). To this end, the

fusion procedure was designed as a global spatio-temporal optimizer minimizing the following energy function:

$$E(r, x) = \sum_{s} V(r_s, x_s), \tag{2}$$

where $V(r_s, x_s)$ is a local energy term and r_s and x_s are assumed to be independent. This energy term returns a low value when the neighborhood surrounding s (called η_s) is spatially and temporally uniform. To measure the *degree of uniformity* of a neighborhood η_s, the local energy term uses two potential functions applied on every site $t \in \eta_s$

$$V(r_s, x_s) = -\sum_{t \in \eta_s} \delta_{r_t, r_s} \delta_{x_t, x_s}. \tag{3}$$

Here, η_s is a square integration window of size $L \times L$ centered on s and δ is the Kronecker delta function. $V(.)$ works in a similar way the well known K-nearest neighbor algorithm does [16]. For a given site s and its neighborhood η_s, $V(r_s, x_s)$ counts the number of sites $t \in \eta_s$ that are simultaneously in spatial region r_s and part of motion class x_s. In this way, the class $x_s \in \Gamma$ that occurs the most often within region η_s is the one with the smallest energy. The way $V(.)$ works is illustrated in Fig.2. In image r, site α is part of the black class (which is a section of the vehicle) but has the *immobile* label in $x^{[0]}$. When looking at every site in η_α *part of the black section of the vehicle in* r, we see there is a majority of sites with *mobile* label in $x^{[0]}$. In other words, within the K-nearest neighbors around site s with a black label in r, there is a majority of *mobile* sites. For this reason, $V(r_\alpha, \text{mobile}) < V(r_\alpha, \text{immobile})$ and thus, α is assigned a *mobile* label in the resulting motion field $x^{[n]}$. The system works in a similar way for site β.

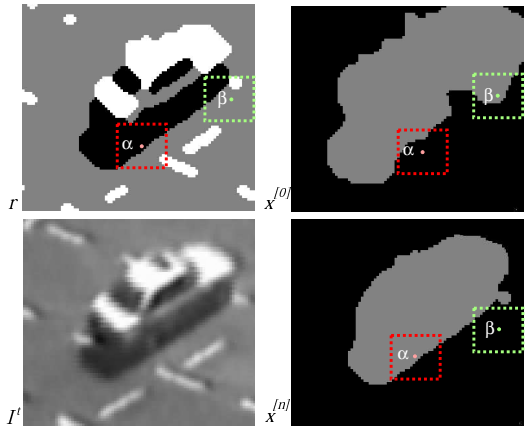

Fig. 2. *Zoom on* Karlsruhe *sequence. Top left is label field* r *and top right is motion label field* $x^{[0]}$*. The motion label field contains two classes which can be understood as the "mobile" and "immobile" classes. Bottom left is the image frame at time* t *while bottom right shows the motion label field after the* n^{th} *iteration. Note how* $x^{[n]}$*'s region silhouette is well localized as compared to* $x^{[0]}$*'s.*

Since there are no analytical solutions to $x = \arg\max_{x'} E(r, x')$, we resort to a classical iterative ICM [2] technique whose mode (the maximum local energy for each site at each iteration) is defined by local energy function $V(r_s, x_s)$. The complete algorithm of our method is presented in Algo. 1.

K-Nearest-Neighbor-Based Fusion Procedure

I^t	Image frame at time t
v	Vector field between I^t and I^{t-1}
r	Spatial segmentation of I^t
$x^{[k]}$	Motion label field after k^{th} iteration
η_s	Window of size $L \times L$ centered at site s
$\delta_{a,b}$	Kronecker delta
m,m'	Number of motion/spatial classes

1. Initialization
$v \leftarrow$ optical flow between I^t and I^{t-1}
$x^{[0]} \leftarrow$ segmentation of v in m classes
$r \leftarrow$ segmentation of image I^t in m' classes
$i \leftarrow 0$

2. ICM Optimization (Fusion)
do
  $i \leftarrow i + 1$
  for *each site* $s \in S$ **do**
    for *each class* $x_c \in \Gamma$ **do**
      $V(r_s, x_c) \leftarrow \sum_{t \in \eta_s} \delta_{r_t, r_s} \delta_{x_c, x_t^{[i-1]}}$
    $x_s^{[i]} \leftarrow \arg\min_{x_c \in \Gamma} V(r_s, x_c)$
  while $x^{[i-1]} \neq x^{[i]}$

Algorithm 1: *Our spatio-temporal motion segmentation algorithm based on a K-nearest neighbor algorithm.*

6 Experimental Results

To validate our algorithm, we have segmented sequences representing different challenges. Some sequences are real while others are synthetic. The latter come with perfect ground-truth image g and with various degrees of difficulty. The tests presented aim at validating how stable and robust our algorithm is with respect to the window size $L \times L$ and to the precision of the spatial partition r.

At first, we built two synthetic sequences with different textures that are more or less easy to segment spatially. As shown in Fig.4, the sequences allow a well defined spatial partition r, a medium and a badly defined partition r. In the badly defined partitions (cf. last column of Fig.3), the objects edges in r are barely recognizable. To measure how precise our algorithm is as compared to ground-truth image g, we have computed the percentage of bad matching pixels [20], i.e.,

$$B = \frac{1}{N_S} \sum_{s \in S} (1 - \delta_{x_s, g_s}) \tag{4}$$

where N_S is the number of sites in S and δ_{x_s, g_s} is the Kronecker delta function.

In Fig.4, we compare our results to the ones obtained with methods close to ours. The first method is Tekalp's MAP [7, 8] which is a motion-based Markovian approach using affine motion models. The results return by this method are visually similar to $x^{[0]}$ (c.f. third row of Fig. 3). The second method is Altunbasak *et al.*'s [13] region-based approach which relies on a pre-estimated segmentation map r. As shown in Fig.4, their method is more sensitive to the precision of r. These results underline the fact that our algorithm reacts smoothly to a change of its parameters L and r. It is thus stable and doesn't generate unexpected results especially when segmented regions in r don't exhibit precise edges.

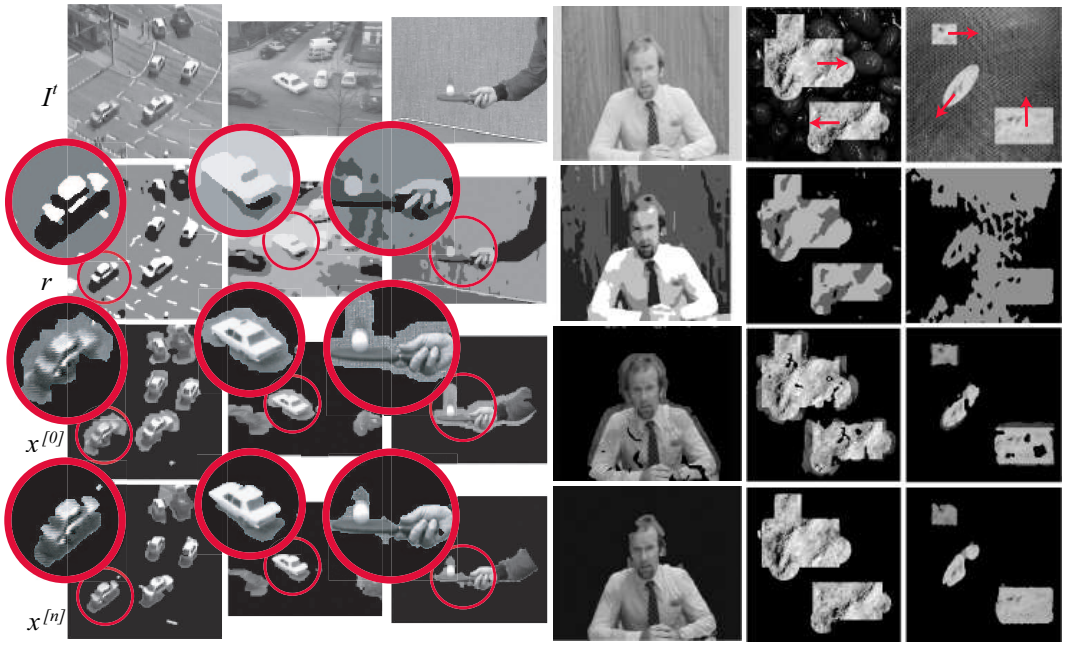

Fig. 3. *Sequences* Karlsruhe, Taxi, Tennis, Trevor White, SequenceA, *and* SequenceB. SequenceA *and* SequenceB *are synthetic sequences with respectively a* precise *and an* imprecise *spatial partition* r. *The first row presents frames at time* t, *the second row spatial partitions* r *and the last two rows the motion label fields* $x^{[0]}$ *and* $x^{[n]}$ *superposed to* I^t. *Notice that* $x^{[0]}$ *is visually similar to the results returned by Tekalp MAP algorithm [7, 8].*

As for the real sequences, we superposed the motion label fields $x^{[0]}$ and $x^{[n]}$ with image I^t to illustrate how precise the results are. Results are shown in Fig.3. From left to right, sequences were segmented with respectively three, three, four, six, four, and three motion classes. We can see that in most cases, the segmentation map returned by our algorithm is more accurate than the ones with no fusion procedure.

Sequence A

Partition r	MAP	Alt.	$x^{[0]}$	3×3	7×7	11×11	21×21	31×31
precise	15.7	0.8	13.2	13.1	5.0	1.9	1.0	0.9
mediocre	12.5	12.5	10.8	10.7	5.4	4.0	4.2	5.3
imprecise	6.0	25.5	8.1	8.1	5.4	5.3	8.3	9.3

Sequence B

Partition r	MAP	Alt.	$x^{[0]}$	3×3	7×7	11×11	21×21	31×31
precise	11.1	0.4	6.2	2.9	0.4	0.4	0.4	0.5
mediocre	11.6	8.9	6.7	3.3	0.7	0.8	0.9	1.3
imprecise	12.4	42.6	5.2	3.3	2.0	2.6	2.7	5.4

Fig. 4. *Percentage of bad matching pixels computed with three different versions of two synthetic image sequences. From left to right: results obtained with Tekalp's MAP algorithm [7, 8], Altunbasak* et al. *[13], our unsupervised statistical Markovian algorithm and results obtained with our fusion algorithm. The five rightmost columns measure the effect of the window size* ($L \times L$). *The quality of the spatial partition* r *is ranked from* precise *to* imprecise *depending on how well objects have been segmented (see second row of Fig.3).*

7 Discussion

In this paper, we have considered the issue of segmenting an image sequence based on spatial and motion cues. The core of our method is a K-nearest-neighbor-based fusion between a spatial partition r and a temporal partition $x^{[0]}$. The two fields are blended together by an ICM optimization procedure that minimizes an energy function made

of a spatio-temporal potential function. This function works in a similar way the K-nearest neighbor algorithm does.

Although a spatio-temporal segmentation based on pre-estimated label fields might appears as a step backward when compared to methods such as Black's [12] or Khan and Shaw's [15] (that minimize one large spatio-temporal energy function) it has legitimate advantages. To start off with, these methods rely heavily on weighting functions and/or on weighting coefficients that give more or less influence to the temporal data vs the spatial data. A bad choice of these parameters can resolve in a bad segmentation. Also, because these parameters generally depend on the sequence content, they have to be re-estimated when used on new sequences. Unfortunately, tweaking these weighting factors isn't trivial, especially when their number is large (such as 8 for Black's [12]). Furthermore, large energy functions (the ones with many energy terms and/or defined over multidimensional data) are generally less stable than smaller ones and thus need sometimes to be implemented along with a stochastic (and slow) optimization procedure such as simulated annealing.

The point with our method is to alleviate these problems by minimizing individually the spatial and temporal energy functions before to blend it together. Our method can thus be seen as a divide-and-conquer approach that doesn't rely on weighting factors. It uses short energy functions that can be minimized with a deterministic optimization procedure which converges faster than stochastic solutions. This makes the solution stable and tractable. Furthermore, we believe our fusion method is trivial to implement and, since it processes every pixels independently, it could be efficiently implemented on parallel hardware.

Results obtained on real and synthetic image sequences shows that our algorithm is stable and precise. It reacts well to a change of its parameters and/or to a poorly estimated spatial label field r. In the future, we look forward to extend this method to other vision problems such as stereovision, motion detection and motion estimation.

References

1. Zhang D. and Lu G. Segmentation of moving objects in image sequence: A review. *Circuits, Systems and Signal Process.*, 20(2):143–183, 2001.
2. Besag J. On the statistical analysis of dirty pictures. *J. Roy. Stat. Soc.*, 48(3):259–302, 1986.
3. Megret R. and DeMenthon D. A survey of spatio-temporal grouping techniques. Technical report, University of Maryland, College Park, 2002.
4. Bouthemy P. and Lalande P. Recovery of moving object masks in an image sequence using local spatiotemporal contextual information. *Optical Engineering*, 32(6):1205–1212, 1993.
5. J.-M. Odobez and P. Bouthemy. Robust multiresolution estimation of parametric motion models. *Journal of Visual Communication and Image Representation*, 6(4):348–365, 1995.
6. Stiller C. and Konrad J. Estimating motion in image sequences: A tutorial on modeling and computation of 2d motion. *IEEE Signal Process. Mag.*, 16:70–91, 1999.
7. A. Murat Tekalp. *Digital video processing*. Prentice-Hall, Inc., 1995.
8. Bovik A., editor. *Handbook of Image and Video Processing*. pub-ACADEMIC, 2000.
9. Murray D. and Buxton B. Scene segmentation from visual motion using global optimization. *IEEE Trans. Pattern Anal. Machine Intell.*, 9(2):220–228, 1987.
10. Darrell T. and Pentland A. Cooperative robust estimation using layers of support. *IEEE Trans. Pattern Anal. Mach. Intell.*, 17(5):474–487, 1995.
11. Wang J. and Adelson E. Representing moving images with layers. *The IEEE Trans. on Image Proces. Sp. Issue: Image Seq. Comp.*, 3(5):625–638, September 1994.
12. Black M. Combining intensity and motion for incremental segmentation and tracking over long image sequences. In *Proc. of the Sec. European Conf. on Compt. Vis.*, pages 485–493, 1992.
13. Altunbasak Y., Eren P., and Tekalp M. Region-based parametric motion segmentation using color information. *Graph. Models Image Process.*, 60(1):13–23, 1998.
14. Bergen L. and Meyer F. A novel approach to depth ordering in monocular image sequences. In *Proc. of CVPR*, pages 536–541, 2000.
15. Khan S. and Shah M. Object based segmentation of video color, motion and spatial information. In *Proc. of CVPR*, pages 746–751, 2001.
16. Bishop C. *Neural Networks for Pattern Recognition*. Oxford University Press, 1996.
17. Pieczynski W. Statistical image segmentation. *Machine Graphics and Vision*, 1(1):261–268, 1992.
18. Bouguet J.-Y. Pyramidal implementation of the lucas kanade feature tracker: Description of the algorithm. Technical report, Intel Corporation, 1999.
19. Simoncelli E., Adelson E., and Heeger D. Probability distributions of optical flow. In *Proc. of CVPR.*, pages 310–315, 1991.
20. Scharstein D., Szeliski R., and Zabih R. A taxonomy and evaluation of dense two-frame stereo correspondence algorithms. In *Proc. of the IEEE Workshop on Stereo and Multi-Baseline Vision*, 2001.