# Detection and segmentation of moving objects in highly dynamic scenes

Aurélie Bugeau, Patrick Pérez

# Detection and segmentation of moving objects in highly dynamic scenes

Aurélie Bugeau Patrick Pérez
INRIA, Centre Rennes - Bretagne Atlantique
Université de Rennes 1, Campus de Beaulieu, 35 042 Rennes Cedex, France
{aurelie.bugeau,perez}@irisa.fr

## Abstract

*Detecting and segmenting moving objects in dynamic scenes is a hard but essential task in a number of applications such as surveillance. Most existing methods only give good results in the case of persistent or slowly changing background, or if both the objects and the background are rigid. In this paper, we propose a new method for direct detection and segmentation of foreground moving objects in the absence of such constraints. First, groups of pixels having similar motion and photometric features are extracted. For this first step only a sub-grid of image pixels is used to reduce computational cost and improve robustness to noise. We introduce the use of p-value to validate optical flow estimates and of automatic bandwidth selection in the mean shift clustering algorithm. In a second stage, segmentation of the object associated to a given cluster is performed in a MAP/MRF framework. Our method is able to handle moving camera and several different motions in the background. Experiments on challenging sequences show the performance of the proposed method and its utility for video analysis in complex scenes.*

## 1. Introduction

Detection of moving objects in sequences is an essential step for video analysis. It is a difficult task in the presence of a dynamic background. Different kinds of methods exist to solve the problem of motion detection and motion segmentation. One of them is background modeling and subtraction, which is a preliminary step to moving object detection and subsequent processing is necessary to get the masks of moving objects. First works were based on adjacent frames difference [12]. However, this simple method is unsuitable for real world situations and statistical methods were introduced to model the background. Background modeling methods can be classified as predictive or non predictive methods. Non-predictive methods build a probability density function of the intensity at an individual pixel [8] [10]. Non parametric approaches are more suited when the density function becomes complex [7]. Until recently, most methods were based on photometric properties. In [15], Mittal and Paragios present a non parametric algorithm that combines color and flow features, and introduce a variable bandwidth kernel. Predictive methods use a dynamical model to predict the value of a pixel from previous observations [25]. All these pixel-wise approaches allow an accurate detection of moving objects but are memory and possibly computationally expensive. Also, they can be sensitive to noise and they don't take into account spatial correlation. For these reasons, spatial consistency can be added as in [19], where a MAP-MRF modeling of both foreground and background is used to detect moving objects. This method has been extended to novelty detection in [14]. Feature based models also exist for background modeling. For example, in [26], the background is modeled only on corners, and moving objects are then found by the clustering of foreground features trajectories. For numerous outdoor sequences, the changes in the background appear suddenly and, in case of grayscale videos, the objects may have intensity values close to the ones of the background. Hence, background modeling is difficult and often not sufficient.

Another approach to detect moving objects is to extract groups of salient motion by accumulating flows consistent in terms of direction over successive frames [23] [20].

Motion segmentation can also be seen as the problem of fitting a collection of motion models to the image data. These layered approaches often use EM algorithm [22] or more recently graph cuts [24] to extract layers. The problem can also be cast in terms of multi-body factorization, and many papers can be found on this subject when the scene is static. In [21], it was adapted for both static and dynamic scenes. Recently, in [17], an incremental approach to layer extraction has been introduced. Feature points are detected, tracked and then merged into groups based on their motion. Objects are detected incrementally when enough evidence can distinguish them from their background.

In this paper, we are interested in challenging sequences containing complex motions, possibly with high amplitude, and sudden changes in the background. For example, in the context of driver surveillance, the motions visible through the windows are often hard to characterize. The "background" is composed of both the passenger compartment and what is behind the windows. Furthermore, contrast between background and interesting objects (face, hands) can be low. Also, the sequences we consider can be shot by a moving camera. Our work does not aim at modeling the background or at finding every layer but only at detecting moving foreground objects. We define these objects as groups of pixels that are salient for both motion and color. Our algorithm can be divided in four main steps. First, the camera motion is computed and the images rectified (section 2). All pixels whose motion is close to the camera motion are left apart for the two next steps. In order to reduce the computational cost and to be more robust to noise, we restrict momentarily the analysis to a subgrid of "moving" pixels, i.e. not belonging to camera motion (section 3). A descriptor is defined to characterize them. They are then merged into clusters consistent for both color and motion (section 4). From the clusters, the complete pixel-wise segmentation of moving objects is found using a MAP-MRF framework (section 5). Finally, section 6 presents some experimental results.

## 2. Sensor motion

Most of the test sequences we are working on have been taken by a moving handheld camera. We assume that the apparent motion induced by the physical motion of the camera is dominant in the image and is well approximated by an affine motion field. In this paper, $I_t^{(g)}$ denotes the grayscale image at time $t$, $I_t^{(c)}$ the color image and $\mathcal{P}$ the set of pixels in the image $I_t^{(g)}$. The displaced frame difference between two consecutive frames $I_{t+1}^{(g)}$ and $I_t^{(g)}$ is given by:

$$D_t(p) \;=\; I_{t+1}^{(g)}(p + \vec{w}_t(p)) - I_t^{(g)}(p) + \zeta_t \quad , \qquad (1)$$

where $p$ is a pixel ($p \in \mathcal{P}$), $\vec{w}_t(p)$ the associated flow vector and $\zeta_t$ a global intensity shift to account for global illumination changes. As in [16], the estimation of the parameters defining motion field $\vec{w}_t$ and global shift $\zeta_t$ is done using an M-estimator. The weight map of the M-estimator is denoted as $W_t$ ($W_t(p) \in [0,1]$). The final map indicates if a pixel participates to the robust motion estimation ($W_t(p)$ close to 1) or is more considered as an outlier ($W_t(p)$ close to 0). A simple pixel-wise motion detector can be built using this map. A pixel is considered as "moving" at time $t$ if it is an outlier to the dominant motion at times $t$ and $t-1$:

$$M_t(p) = \begin{cases} 1 \text{ if } W_t(p + \vec{w}_{t-1}(p)) + W_{t-1}(p) \;=\; 0 \\ 0 \text{ else } . \end{cases} \qquad (2)$$

If, for a pixel $p$, $M_t(p) = 0$, it is considered as a motionless pixel. In the sequel, $\tilde{I}_{t+1}^{(g)}$ will denote back-warped images:

$$\tilde{I}_{t+1}^{(g)}(p) \;=\; I_{t+1}^{(g)}(p + \vec{w}_t(p)) + \zeta_t.$$

## 3. Selection and description of points

The goal of the algorithm is to build and segment groups of pixels consistent both for motion and for some photometric or colorimetric features. These groups must correspond to interesting moving objects. Processing is only done on a subset of moving points and their neighborhoods. This section presents the definition of this subset of points and the point description used to perform clustering.

### 3.1. Selection

In [26], the authors have chosen to use corners, detected with the Harris corner detector. The authors justify the use of corners by claiming that a moving object contains a large number of corners. In our experiments, we have observed that the number of corners belonging to a moving object can be much lower than the number of corners belonging to the background. Besides, if variations in the background are fast and if parallax changes, the number of corners and their neighborhood can be signicatively different from one frame to the other. Finally, corner detection adds one stage of calculation and requires two thresholds.

As no a priori is assumed about the shape and texture of objects, we have chosen to use points of arbitrary type. Hence, we only use a grid of points regularly spread on the image. As the purpose is to detect moving objects, the simple pixel-wise motion detector from section 2 is used to restrict this step to the grid subset:

$$\mathcal{G} = \{p = (\frac{k.w}{N}, \frac{l.h}{N}), k = 0 \cdots N \,,\, l = 0 \cdots N \mid M_t(p) = 1\} \;, \qquad (3)$$

where $w$ and $h$ are the dimensions of the image and $N^2$ the size of the grid before pruning. The value of the parameter $N$ is important. It controls the balance between computational cost (regional methods) and accuracy (local methods). Next step of the algorithm can become computationally expensive if the number of points of the grid is too large. An important thing to note is that $N$ may depend on the number $m$ of "moving" pixels in the image, $m = \sum_{p \in \mathcal{P}} M_t(p)$. To limit the computational cost for clusters creation, we fix the number of points $n$ (1000 in our experiments) that will be kept in further steps of the algorithm. The size $N$ of the grid is then set as $N = \sqrt{w * h * n/m}$.

### 3.2. Description

Now that the points are chosen, the features that will be used to create clusters corresponding to objects need to be defined. It is necessary to chose only few discriminant features. An object is defined as a moving and compact area over which the values of displacement and photometry are nearly constants. Color is not sufficient because the contrast between an object and the background can be small,

as is flow in case of similar motion between an object and the background. Hence the descriptor is formed by three different groups of features. The first group is composed of the coordinates of the point. The second group contains its motion, and the last one contains discriminant photometric features.

#### 3.2.1 Motion features

As we try to detect moving objects, an essential feature is the displacement of the selected points. It is computed using an optical flow technique robust to local linear illumination changes. We used Lucas and Kanade algorithm [13], with an incremental multiscale implementation. A parameter $a$ that accounts for local illumination changes has been added. The flow $(d_x, d_y)$ at each particular point $p = (x, y)$ of the grid is then obtained by solving:

$$\text{argmin}_{a,d_x,d_y} \sum_{(x',y')\in\mathcal{V}(p)} (a\tilde{I}^{(g)}_{t+1}(x'+d_x, y'+d_y) - I^{(g)}_t(x', y'))^2 \tag{4}$$

where $\mathcal{V}(p)$ is the neighborhood of $p$. As it is well known, Lucas and Kanade algorithm has some drawbacks: the brightness constancy is not satisfied and there is no spatial consistency. We could have used Horn and Schunk algorithm [11] that adds a smoothness term to regularize over the whole image or the robust estimation of Black and Anandan [1] to get a better estimation. However these algorithms are more expensive and we do not aim at having a perfect estimation over the all image.

To validate values of displacement, a comparison is done between the neighborhood of pixel $p = (x, y)$ in image at time $t$ (data sample $X$), and the neighborhood of point $p' = (x + d_x, y + d_y)$ at time $t + 1$ (data sample $Y$). The linear relationship between intensity values of $X$ and $Y$ is estimated by computing the normalized cross correlation $r$. Unfortunately, the correlation does not take into account the individual distributions of $X$ and $Y$. Hence it is a poor statistics for deciding whether or not two distributions are really correlated. Statistical tests exist to assess this correlation. One of such tests is based on so-called "p-value". The p-value is the probability that the results have been obtained by chance alone. Here the null hypothesis asserts that the two distributions are uncorrelated. If one wants to limit to $5\%$ the risk that a false positive error has occurred, then data are assumed correlated if the p-value is lower than $0.05$. If not, the motion vector at point $p$ is considered as a non valid and will not be used in next steps of the algorithm. Finally, a new grid

$$\mathcal{G} = \{p = (\frac{k.w}{N}, \frac{l.h}{N}) \mid M_t(p) = 1 \ \& \ \text{pvalue}(p, p') < 0.05\} \tag{5}$$

is obtained with a flow vector $F(p)$ associated to each of its point $p$. The size of the grid $\mathcal{G}$ will be denoted as $M = |\mathcal{G}|$.

#### 3.2.2 Photometric features

To be robust to noise, the photometric features are computed over the neighborhood of each point of the grid defined in previous subsection. We observed that the three RGB color channels do not give the best representation of images. In fact most of our test sequences contain human skin, which has a specific signature in the space of chrominance. Hence, it is interesting to use instead a color system representing the chrominance, *e.g.*, the system YUV. This choice proved appropriate for various types of sequences. To include some simple temporal consistency, we add image $t + 1$ chrominance values of the corresponding point.

Finally, the descriptor at each individual valid point indexed by $i$ ($i = 1 \dots M$) of the grid is:

$$\mathbf{x}^{(i)} = \{\mathbf{x}^{(i)}_1, \mathbf{x}^{(i)}_2, \mathbf{x}^{(i)}_3\}, \tag{6}$$

where
$\mathbf{x}^{(i)}_1 = \{x, y\}$ , $\mathbf{x}^{(i)}_2 = \{d_x, d_y\}$ ,
$\mathbf{x}^{(i)}_3 = \{\overline{Y_t}(x, y), \overline{U_t}(x, y), \overline{V_t}(x, y),$
$\overline{Y_{t+1}}(x', y'), \overline{U_{t+1}}(x', y'), \overline{V_{t+1}}(x', y')\},$
with $(x', y') = (x + d_x, y + d_y)$, and $\bar{\cdot}$ denotes the mean over the neighborhood.

## 4. Grouping points

Now that a grid of valid points has been chosen and described, we address the problem of grouping the points into clusters.

### 4.1. Mean shift for mixed feature spaces

An appealing technique to extract the clusters is the Mean Shift algorithm, which does not require to fix the (maximum) number of clusters. On the other hand the kernel bandwidth and shape for each dimension has to be chosen or estimated. Mean shift is an iterative gradient ascent method used to locate the density modes of a cloud of points, i.e. the local maxima of its density [6]. Here the theory is briefly reminded. Given the set of points $\{\mathbf{x}^{(i)}\}_{i=1..M}$ in the $d$-dimensional space $\mathbb{R}^d$, the non-parametric density estimation at each point $\mathbf{x}$ is given by:

$$\widehat{f}_{\mathbf{H},k}(\mathbf{x}) = \frac{1}{n(2\pi)^{d/2}\|\mathbf{H}\|^{1/2}} \sum_{i=1}^{M} k(\|\mathbf{H}^{-1/2}(\mathbf{x} - \mathbf{x}^{(i)})\|^2) \tag{7}$$

where $k$ is a kernel profile and $\mathbf{H}$ the bandwidth matrix. Introducing the notation

$$g(\mathbf{x}) = -k'(\mathbf{x})$$

leads to the density gradient :

$$\nabla\widehat{f}_{\mathbf{H},k}(\mathbf{x}) = \mathbf{H}^{-1}\, \widehat{f}_{\mathbf{H},g}(\mathbf{x})\, \mathbf{m}_{\mathbf{H},g}(\mathbf{x}) \tag{8}$$

where $\mathbf{m}_{\mathbf{H},g}$ is the "mean shift" vector,

$$\mathbf{m}_{\mathbf{H},g}(\mathbf{x}) = \frac{\sum_{i=1}^{M} \mathbf{x}^{(i)}\, g\left(\|\mathbf{H}^{-1/2}(\mathbf{x}-\mathbf{x}^{(i)})\|^2\right)}{\sum_{i=1}^{M} g\left(\|\mathbf{H}^{-1/2}(\mathbf{x}-\mathbf{x}^{(i)})\|^2\right)} - \mathbf{x} \quad . \quad (9)$$

Using exactly this displacement vector at each step guaranties convergence to the local maximum of the density. With a $d$-variate Gaussian kernel, equation 9 becomes

$$\mathbf{m}_{\mathbf{H},g}(\mathbf{x}) = \frac{\sum_{i=1}^{M} \mathbf{x}^{(i)}\, \exp(-\frac{1}{2}D^2(\mathbf{x},\mathbf{x}^{(i)};\mathbf{H}))}{\sum_{i=1}^{M} \exp(-\frac{1}{2}D^2(\mathbf{x},\mathbf{x}^{(i)};\mathbf{H}))} - \mathbf{x} \quad (10)$$

where

$$D^2(\mathbf{x},\mathbf{x}^{(i)};\mathbf{H}) \equiv (\mathbf{x}-\mathbf{x}^{(i)})^T\mathbf{H}^{-1}(\mathbf{x}-\mathbf{x}^{(i)}) \quad (11)$$

is the Mahalanobis distance from $\mathbf{x}$ to $\mathbf{x}^{(i)}$.

Assume now that the $d$-dimensional space can be decomposed as the Cartesian product of $S$ (3 in our case) independent spaces associated to different types of information (*e.g.* position, color ...), also called feature spaces or domains, with dimensions $d_s, s = 1 \dots S$ (where $\sum_{s=1}^{S} d_s = d$). Because the different types of information are independent, the bandwidth matrix $\mathbf{H}$ becomes $\mathbf{H} = \mathrm{diag}[\mathbf{H}_1 \dots \mathbf{H}_S]$ and thus the mean shift vector can be rewritten as

$$\mathbf{m}_{\mathbf{H},g}(\mathbf{x}) = \frac{\sum_{i=1}^{M} \mathbf{x}^{(i)} \prod_{s=1}^{S} \exp(-\frac{1}{2}D^2(\mathbf{x}_s,\mathbf{x}_s^{(i)};\mathbf{H}_s))}{\sum_{i=1}^{M} \prod_{s=1}^{S} \exp(-\frac{1}{2}D^2(\mathbf{x}_s,\mathbf{x}_s^{(i)};\mathbf{H}_s))} - \mathbf{x} \quad (12)$$

where $\mathbf{x}^{(i)T} = (\mathbf{x}_1^{(i)T}, \dots, \mathbf{x}_S^{(i)T})$ and $\mathbf{x}^T = (\mathbf{x}_1^T, \dots, \mathbf{x}_S^T)$. The mean shift filtering is obtained by successive computations of equation 10 or 12 and translation of the kernel according to the mean shift vector. This procedure converges to the local mode of the density [6].

### 4.2. Bandwidth selection

The partition of the feature space is obtained by grouping together all the data points whose associated mean shift procedures converged to the same mode. The quality of the results highly depends on the choice of the bandwidth matrix $\mathbf{H}$. In [5], Comaniciu proposes to find the best bandwidths within a range of $B$ predefined matrices $\{\mathbf{H}^{(b)}, b = 1 \dots B\}$. Mean Shift partitioning is first run at each scale (for $b$ varying from 1 to $B$). For each data point $\mathbf{x}^{(i)}$, an analysis of the sequence of clusters to which the point is associated is performed. The scale for which the cluster is the most stable is selected, along with associated bandwidth, for data point $\mathbf{x}^{(i)}$. Therefore, the algorithm can be decomposed in two steps. The first one is called bandwidth evaluation at the partition level. It consists in finding a parametric representation of each cluster in order to do the comparisons. The second step called evaluation at the data level is the analysis of cluster sequences at each data point.

An iterative algorithm dedicated to bandwidth selection for mixed feature spaces has been derived from this method [4]. Best bandwidths are then iteratively found for position, color and motion. The range of predefined matrices for color and motion is directly computed from image noises. Introducing $\mathcal{C}$ the set of pairs of neighboring points of the grid, $|\mathcal{C}|$ its cardinal, $\mathbf{I}_{d_s}$ the identity matrix of dimension $d_s$, and the mean and standard deviation :

$$\alpha_s = \frac{1}{|\mathcal{C}|} \sum_{(i,j)\in\mathcal{C}} |\mathbf{x}_s^{(i)} - \mathbf{x}_s^{(j)}|, \quad (13)$$

$$\beta_s = \sqrt{\frac{1}{|\mathcal{C}|} \sum_{(i,j)\in\mathcal{C}} (|\mathbf{x}_s^{(i)} - \mathbf{x}_s^{(j)}| - \alpha_s)^2}\ , \quad (14)$$

the range of matrices for color ($s = 3$) and motion ($s = 2$) of size $d_s$ can be written as

$$\mathbf{H}_s^{(b)} = (\alpha + \frac{2b\beta}{B})\mathbf{I}_{d_s}, b = 1 \dots B. \quad (15)$$

The range of matrices for position reads:

$$\mathbf{H}_1^{(b)} = \frac{4b}{B}(\frac{w}{N}, \frac{h}{N})\mathbf{I}_{d_1}, b = 1 \dots B. \quad (16)$$

The best bandwidth obtained at the end of the bandwidth selection algorithm will be denoted as $\tilde{\mathbf{H}} = \mathrm{diag}[\tilde{\mathbf{H}}_1, \tilde{\mathbf{H}}_2, \tilde{\mathbf{H}}_3$ in the sequel.

At the end of the mean shift clustering procedure [4] several clusters are obtained, each corresponding to a moving object or object part. We retain only large enough clusters (*e.g.*, with more than 15 grid points).

## 5. Segmentation

Segmenting the object associated to a given cluster amounts to assigning a label $l_p$, either "background" or "object", to each pixel $p$ of the image. This problem can be reformulated into the graph cut framework as a bi-partitioning problem. Recently graph cuts have been increasingly used in image segmentation. The reason for such a popularity is that the exact maximum a posteriori (MAP) of a two label pairwise Markov Random Field (MRF) can be computed in polynomial time using min-cut/max-flow algorithms [9]. In seminal paper [3], Boykov *et al.* introduce an iterative foreground/background segmentation system based on this principle, using hard constraints provided by the user. Here we can directly learn some properties of the object from the grid points belonging to its cluster. These points are called inliers. The energy function to minimize is defined as:

$$E_t(L) = -\gamma_c \sum_{p\in\mathcal{P}} \ln(\Pr(I_t^{(c)}(p)|l_p)) - \gamma_m \sum_{p\in\mathcal{G}} \ln(\Pr(F(p)|l_p)) + \lambda \sum_{(p,q)\in\mathcal{V}} \exp^{-\frac{\|I_t^{(g)}(p) - I_t^{(g)}(q)\|^2}{\sigma^2}} \cdot \frac{1}{\mathrm{dist}(p,q)}(1-\delta(l_p,l_q)) \quad (17)$$

where $L$ is the set of all the labels $l_p$, $p \in \mathcal{P}$, $\mathcal{V}$ is the set of unordered pairs $(p,q)$ of neighboring elements of $\mathcal{P}$ and $I_t^{(c)}$ is the original RGB color image converted to YUV color space. The parameters $\gamma_m$, $\gamma_c$, $\lambda$ are some weight constants discussed below.

The two first terms of the cost function are based on pixel-wise modeling of color and motion features distributions. Motion term only concerns the points of the grid. For both color and motion, object distributions are built from histograms on the inliers. For the background, histograms are built as follows. For color it is computed on the all image whereas for motion it is only computed on the grid. In [2], authors have shown that it is possible to force some pixels to belong to the object or to the background. Here we force inliers to belong to the object. Because for motion we only take points of the grid, we chose to set the parameters $\gamma_c$ and $\gamma_m$ such that $\gamma_c = 1$ and

$$\gamma_m = \frac{w+h}{2N}. \tag{18}$$

The parameter $\sigma$ in the third energy term can be related to noise [18]. Here we already have its approximate value from the bandwidth selection in mean shift clustering. Thus we chose $\sigma$ as

$$\sigma^2 = \|\tilde{\mathbf{H}}_3\|^2 \ . \tag{19}$$

The value of parameter $\lambda$ has not been really studied in literature. To avoid a possible saturation of all binary edges in the max-flow procedure, we fix here its value as:

$$\lambda = \text{argmin}_p \gamma_c \sum_{p\in\mathcal{P}} \ln(\Pr(I_t^{(c)}(p)|l_p)) + \gamma_m \sum_{p\in\mathcal{G}} \ln(\Pr(F(p)|l_p)). \tag{20}$$

At the end, we obtain one segmentation for each cluster.

## 6. Results of objects detection

Existing methods for motion detection are limited to small or regular motion in the background, to small motion of the objects, or to rigid layers. To demonstrate the strength of our method we show results on three challenging sequences for which these constraints do not necessarily hold.

In figures 1-3, the first column shows several frames of the video sequences. The second and third columns display, overlaid on each of these frames, the results of the mean shift clustering algorithm and of the segmentation algorithm respectively. Different colors are used to represent the different moving objects of the scene. Note that there is no temporal consistency either between objects or between their colors. The assigned colors only depend on the order in which our algorithm detects the objects.

The first video (figure 1) is a tennis sequence which includes a complex background motion within the spectators, the rapid motion of the player and his racket, and the fast pan and zoom-out of the camera. Despite this complex dynamic content, our algorithm detected the player in each frame of the sequence. On the first frame presented here, the racket and the body have a completely different motion and therefore they are detected separately.

The second results (figure 2) are on a sequence of a water skier. The dynamic content of water regions is all the more complex since they include projections behind the skier. Good results on this video are partly due to the use of p-value for the validation of optical flows. Note however that part of the water is sometimes detected as a moving object.

The last sequence presented here (figure 3) shows a person driving a car. This type of sequences is very difficult as various complex motions appear through the window, with sudden speed, illumination and parallax changes. Our algorithm was nonetheless able to capture interesting foreground objects, i.e., the face and the hands, when they were moving. In the second frame, the face stopped moving and therefore is not detected. As with portions of water in the previous example, objects behind the window are sometimes detected by the mean shift clustering algorithm. We believe that adding temporal consistency or tracking would allow the rejection of such transient detections while locking on interesting objects even if they stop moving. Note also that some inliers from the grid remain isolated after graphcut segmentation (such points, hardly visible in the final transparent overlay, can be seen on close-ups). They could be easily eliminated in a post-processing step (*e.g.*, retaining only largest connected components), as often done in static image segmentation.

## 7. Conclusion and future work

We have presented a technique to detect and segment moving objects in complex dynamic scenes shot by possibly moving cameras. As we only work on a sub grid of pixels, and because we do not model the background, this method is not computationally and memory expensive. The use of spatial, dynamic and photometric features allows the extraction of moving foreground objects even in presence of illumination changes and fast variations in the background. Distinctive ingredients of our approach include the use of p-value to validate optical flow vectors, the use of automatic multidimensional bandwidth selection in the mean shift clustering algorithm and the use of sparse motion data in a MAP-MRF framework. It is worth emphasizing that the parameters involved in the preliminary motion computations (optic flow and parametric dominant motion) are fixed to the same values in all experiments, while the other parameters (for clustering and segmentation) are automatically selected. We plan in the future to add temporal consistency either on a frame-to-frame basis or within a tracker whose (re)initialization would rely on detection maps.

## References

[1] M. Black and P. Anandan. A framework for the robust estimation of optical flow. *Proc. Int. Conf. Computer Vision*, 1993. 3

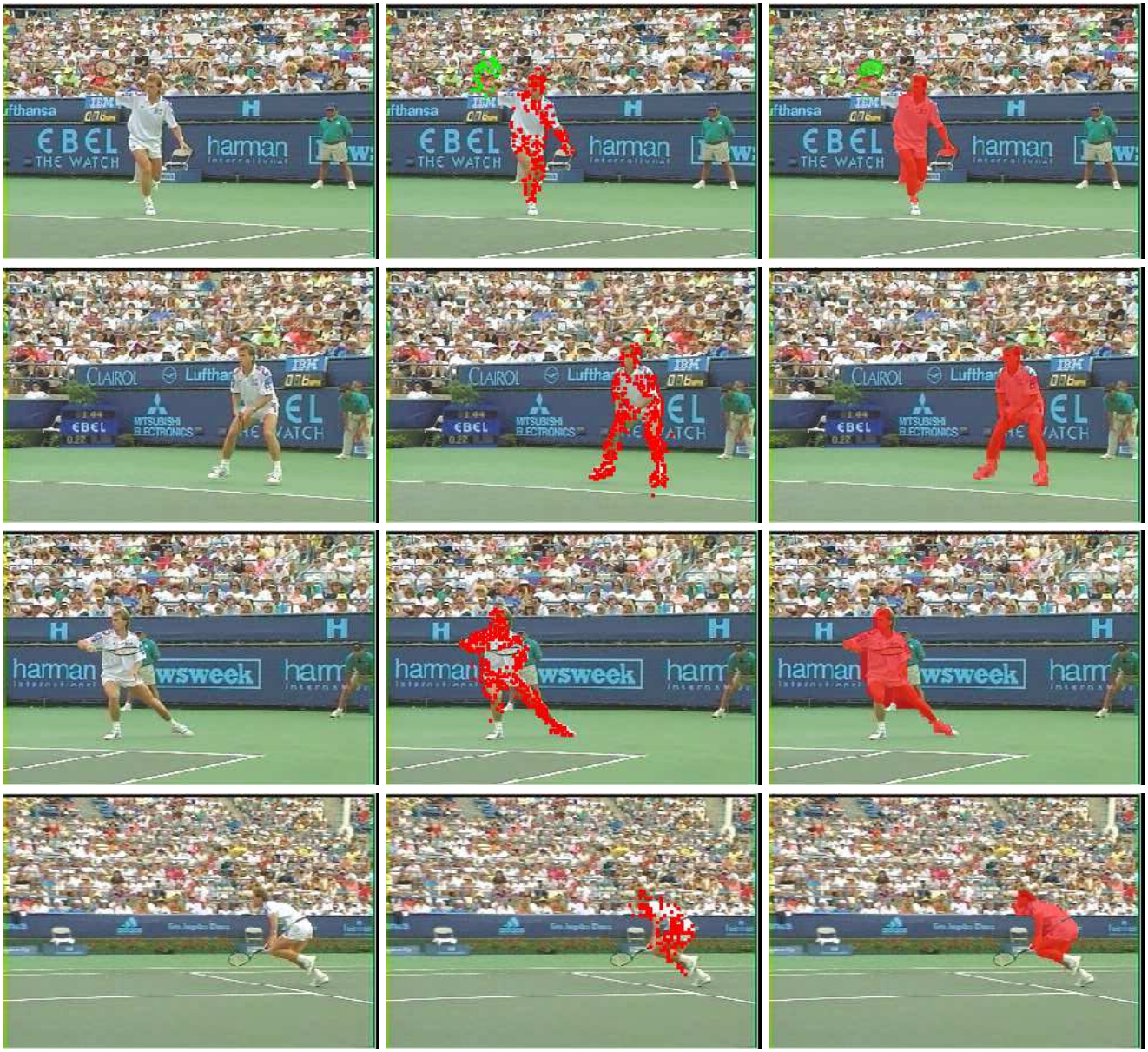


Figure 1. Tennis sequence. Frames 31, 161, 212, 260. See text for details.

[2] Y. Boykov and M. Jolly. Interactive graph cuts for optimal boundary and region segmentation of objects in n-d images. *Proc. Int. Conf. Computer Vision*, 2001. 5

[3] Y. Boykov, O. Veksler, and R. Zabih. Fast approximate energy minimization via graph cuts. *IEEE Trans. Pattern Anal. Machine Intell.*, 23(11):1222–1239, 2001. 4

[4] A. Bugeau and P. Perez. Detection and segmentation of moving objects in highly dynamic scenes. *Technical report, IRISA*, (PI 1846), 2007. 4

[5] D. Comaniciu. An algorithm for data-driven bandwidth selection. *IEEE Trans. Pattern Anal. Machine Intell.*, 25(2):281–288, 2003. 4

[6] D. Comaniciu and P. Meer. Mean shift: A robust approach toward feature space analysis. *IEEE Trans. Pattern Anal. Machine Intell.*, 24(5):603–619, 2002. 3, 4

[7] A. Elgammal, D. Harwood, and L. Davis. Non-parametric model for background subtraction. *Proc. Europ. Conf. Computer Vision*, 2000. 1

[8] N. Friedman and S. Russell. Image segmentation in video sequences: A probabilistic approach. *Uncertainty in Artificial Intelligence*, pages 175–181, 1997. 1

[9] D. Greig, B. Porteous, and A. Seheult. Exact maximum a posteriori estimation for binary images. *J. Royal Statist. Soc.*, 51(2):271–279, 1989. 4

[10] Y. Grimson, C. Stauffer, R. Romano, and L. Lee. Using adaptive tracking to classify and monitor activities in a site. *Proc.*

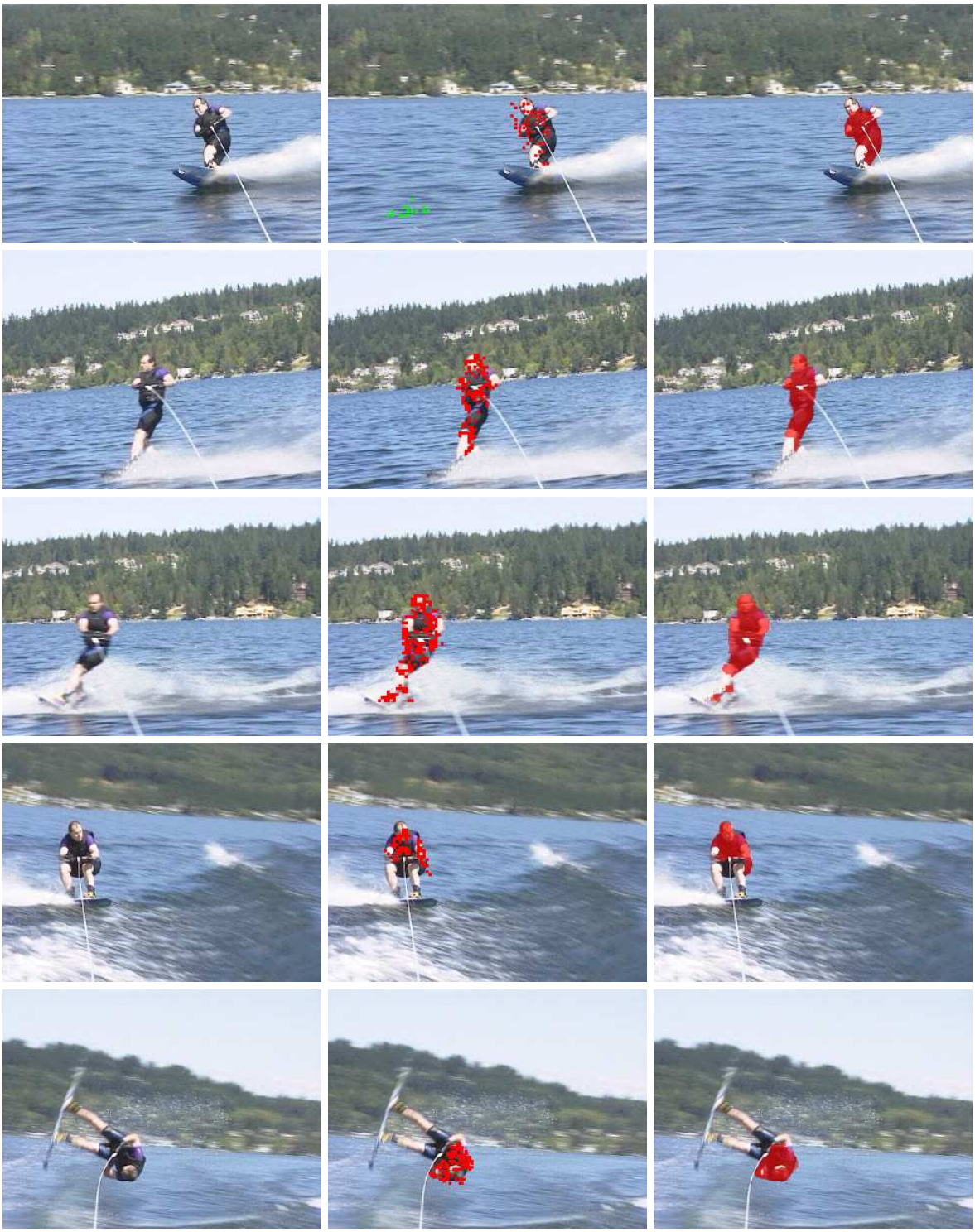

Figure 2. Water skier sequence. Frames 38, 108, 159, 214, 236. See text for details.

*Conf. Comp. Vision Pattern Rec.*, 1998. 1

[11] B. Horn and B. Schunck. Determining optical flow. *Artif. Intell.*, 17(1-3):185–203, 1981. 3

[12] R. Jain and H. Nagel. On the analysis of accumulative difference pictures from image sequence of real world scenes. *IEEE Trans. Pattern Anal. Machine Intell.*, 1(2), 1979. 1

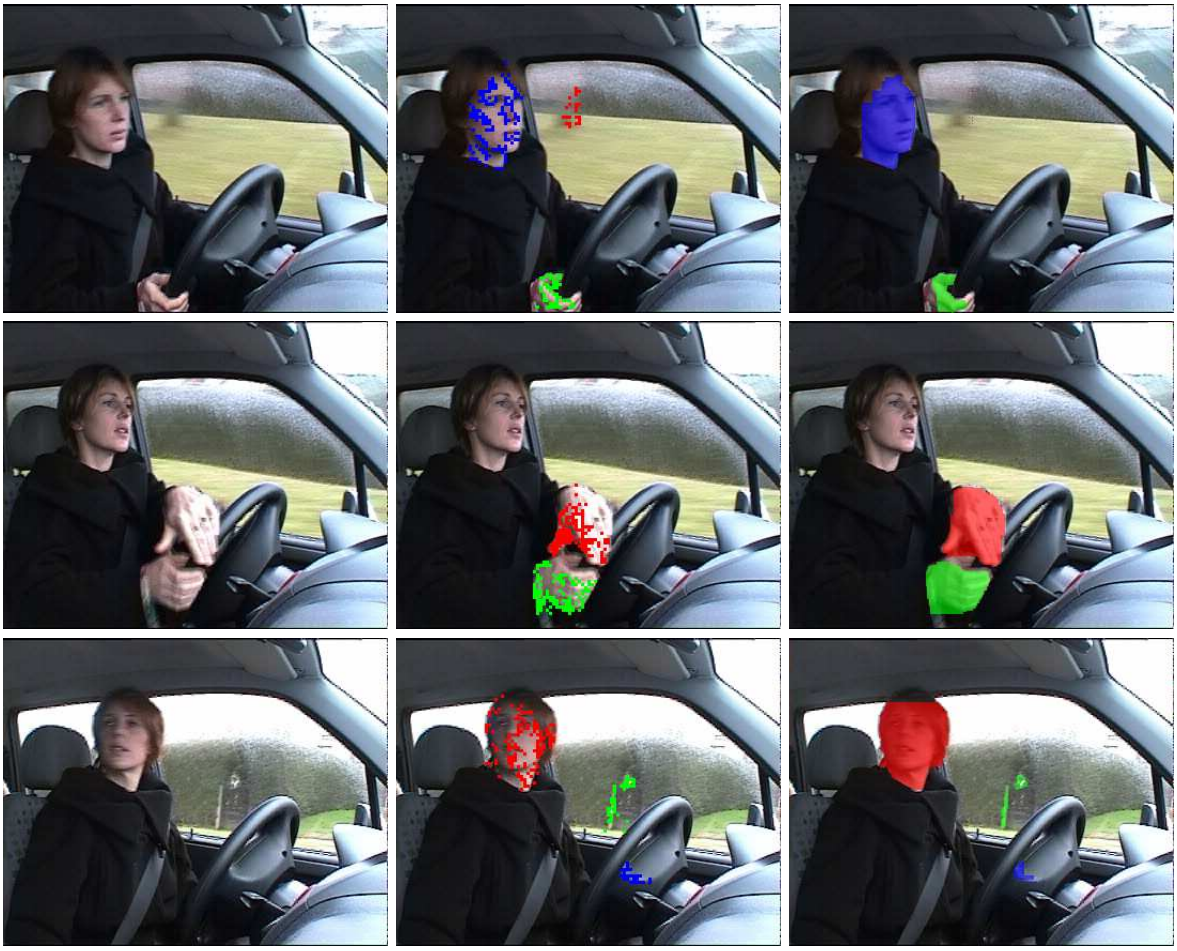

Figure 3. Car driver sequence. Frames 16, 41, 72. See text for details.


[13] B. Lucas and T. Kanade. An iterative technique of image registration and its application to stereo. *Proc. Int. Joint Conf. on Artificial Intelligence*, 1981. 3

[14] S. Mahamud. Comparing belief propagation and graph cuts for novelty detection. *Proc. Conf. Comp. Vision Pattern Rec.*, 2006. 1

[15] A. Mittal and N. Paragios. Motion-based background subtraction using adaptative kernel density estimation. *Proc. Conf. Comp. Vision Pattern Rec.*, 2004. 1

[16] J.-M. Odobez and P. Bouthemy. Robust multiresolution estimation of parametric motion models. *J. Visual Com. and Image Representation*, 6(4), 1995. 2

[17] S. Pundlik and S. Birchfield. Motion segmentation at any speed. *Proc. of the British Machine Vision Conf.*, 2006. 1

[18] C. Rother, V. Kolmogorov, and A. Blake. "grabcut": interactive foreground extraction using iterated graph cuts. *ACM Trans. Graph.*, 23(3):309–314, 2004. 5

[19] Y. Sheikh and M. Sha h. Bayesian modeling of dynamic scenes for object detection. *IEEE Trans. Pattern Anal. Machine Intell.*, 27(11):603–619, 2005. 1

[20] Y. Tian and A. Hampapur. Robust salient motion detection with complex background for real-time video surveillance. *Workshop on Motion and Video Computing*, 2005. 1

[21] R. Vidal and D. Singaraju. A closed form solution to direct motion segmentation. *Proc. Conf. Comp. Vision Pattern Rec.*, 2005. 1

[22] J. Y. A. Wang and E. H. Adelson. Representing moving images with layers. *IEEE Trans. on Image Processing Special Issue*, 3(5):625–638, 1994. 1

[23] L. Wixson. Detecting salient motion by accumulating directionally-consistent flow. *IEEE Trans. Pattern Anal. Machine Intell.*, 22(8):774–780, 2000. 1

[24] J. Xiao and M. Shah. Accurate motion layer segmentation and matting. *Proc. Conf. Comp. Vision Pattern Rec.*, 2005. 1

[25] J. Zhong and S. Sclaroff. Segmenting foreground objects from a dynamic textured background via a robust Kalman filter. *Proc. Int. Conf. Computer Vision*, 2003. 1

[26] S. Zhu, Q. Avidan and K.-T. Cheng. Learning a sparse, corner-based representation for time-varying background modeling. *Proc. Int. Conf. Computer Vision*, 2005. 1, 2