A robust and computationally efficient motion detection algorithm based on Σ-Δ background estimation.

A . Manzanera J. C. Richefeu
ENSTA/LEI
32 Bd Victor
F-75739 PARIS CEDEX 15
http://www.ensta.fr/∼manzaner

Abstract

This paper presents a new algorithm to detect moving objects within a scene acquired by a stationary camera. A simple recursive non linear operator, the Σ-Δ filter, is used to estimate two orders of temporal statistics for every pixel of the image. The output data provide a scene characterization allowing a simple and efficient pixel-level change detection framework. For a more suitable detection, exploiting spatial correlation in these data is necessary. We use them as a multiple observation field in a Markov model, leading to a spatiotemporal regularization of the pixel-level solution. This method yields a good trade-off in terms of robustness and accuracy, with a minimal cost in memory and a low computational complexity.

1. Introduction

To detect the moving objects in an image sequence is a very important low-level task for many computer vision applications, such as video surveillance, traffic monitoring, sign language recognition. When the camera is stationary, a class of methods usually employed is *background subtraction*. The principle of these methods is to build a model of the static scene (i.e. without moving objects) called *background*, and then to compare every frame of the sequence to this background in order to discriminate the regions of unusual motion, called *foreground* (the moving objects).

Many algorithms have been developed for background subtraction: recent reviews and evaluations can be found in [8] [2] [3] [12]. In this paper, we are more specifically interested in outdoor video surveillance systems with long autonomy. The difficulty in devising background subtraction algorithms in such context lies in the respect of several constraints:

- The system must keep working without human interaction for a long time, and then take into account gradual or sudden changes such as illumination variation or new static objects settling in the scene. This means that the background must be *temporally adaptive*.
- The system must be able to discard irrelevant motion such as waving bushes or flowing water. It should also be robust to slight oscillations of the camera. This means that there must be a *local* estimation for the *confidence* in the background value.
- The system must be real-time, compact and low-power, so the algorithms must not use much resource, in terms of computing power and memory.

The two first conditions imply that statistical measures on the temporal activity must be locally available in every pixel, and constantly updated. This excludes any basic approach like using a single model such as the previous frame or a temporal average for the background, and global thresholding for decision.

Some background estimation methods are based on the analysis of the histogram of the values taken by each pixel within a fixed number K of past frames. The mean, the median or the mode of the histogram can be chosen to set the background value, and the foreground can be discriminated by comparing the difference between the current frame and the background with the histogram variance. More sophisticated techniques are also based on the K past frames history: linear prediction [14], kernel density estimation [4] [10], or principal component analysis [11]. These methods require a great amount of memory, since K needs to be large (usually more than 50) for robustness purposes. So they are not compatible with our third condition.

Much more attractive for our requirements are the *recursive* methods, that do not keep in memory a histogram for each pixel, but rather a fixed number of estimates computed recursively. These estimates can be the mean and variance of a Gaussian distribution [15], or different states of the background (e.g. its values and temporal derivatives) estimated by predictive (e.g. Kalman) filter [5]. But it is difficult to get robust estimates of the background with linear

recursive framework, unless a multi-modal distribution (e.g. multiple Gaussian [13]) is explicitly used, which is done at the price of an increasing complexity and memory requirement. Furthermore, these methods rely on parameters such as the learning rates used in the recursive linear filters, setting the relative weights of the background states and the new observations, whose tuning can be tricky, which makes difficult the fulfillment of the first condition stated above.

A recursive approximation of the temporal median was proposed in [9] to compute the background. The interest of this method lies in the robustness provided by the non linearity compared to the linear recursive average, and in the very low computational cost. In this paper, we investigate some nice properties of this method, introducing the notion of Σ-Δ *filtering*, and using it to obtain a non-parametric motion detection. In Section 2, we use the Σ-Δ filter to compute two orders of temporal statistics for each pixel of the sequence, providing a multiple observation field, and a pixel-level decision framework. Then, in Section 3, we exploit the spatial correlation in these data using a Markov based spatial regularization algorithm. High level processing and feedback are then discussed in Section 4, where some results are displayed.

2. Temporal processing

Our first background estimate, shown on Table 1(1), is the same as [9], where I_t is the input sequence, and M_t the estimated background value. As noticed in [9], if I_t is a discrete random signal, the most probable values of M_t lie in an interval $[a, b]$ such that there are as many indices $\tau < t$ such that $I_\tau < a$, as indices $\tau < t$ such that $I_\tau > b$. So M_t is an approximation of the median of I_t. But this filter has other interesting properties, relative to the change detection in time-varying signals. Indeed, we interpret this background estimation as the simulation of a digital conversion of a time-varying analog signal using Σ-Δ modulation (A/D conversion using only comparison and elementary increment/decrement, hence the name Σ-Δ filter).

As the precision of the Σ-Δ modulation is limited to signals with absolute time-derivative less than unity, the modulation error is proportional to the variation rate of the signal, corresponding here to a motion likelihood measure of the pixels. We then use the absolute difference between I_t and M_t as the first *observation* field: the difference Δ_t (Table 1(2)).

Unlike [9], we also use this filter to compute the time-variance of the pixels, representing their motion activity measure, used to decide whether the pixel is more likely "moving" or "stationary". Then the second *observation* field V_t (Table 1(3)) used in our method has the dimension of a temporal standard deviation. It is computed as a Σ-Δ filter of the difference sequence Δ_t. This provides a measure of *temporal activity* of the pixels. As we are interested in pixels whose variation rate is significantly over its temporal activity, we apply the Σ-Δ filter to the sequence of N times the non-zero differences.

Finally, the pixel-level detection is simply performed by comparing Δ_t and V_t (Table 1(4)).

Initialization
for each pixel x:
$M_0(x) = I_0(x)$
For each frame t
for each pixel x:
if $M_{t-1}(x) < I_t(x)$, $M_t(x) = M_{t-1}(x) + 1$
if $M_{t-1}(x) > I_t(x)$, $M_t(x) = M_{t-1}(x) - 1$

(1)

For each frame t
for each pixel x:
$\Delta_t(x) = |M_t(x) - I_t(x)|$

(2)

Initialization
for each pixel x:
$V_0(x) = \Delta_0(x)$
For each frame t
for each pixel x such that $\Delta_t(x) \neq 0$:
if $V_{t-1}(x) < N \times \Delta_t(x)$, $V_t(x) = V_{t-1}(x) + 1$
if $V_{t-1}(x) > N \times \Delta_t(x)$, $V_t(x) = V_{t-1}(x) - 1$

(3)

For each frame t
for each pixel x:
if $\Delta_t(x) < V_t(x)$
then $D_t(x) = 0$
else $D_t(x) = 1$

(4)

Table 1: The Σ-Δ background estimation: (1) Computation of the Σ-Δ mean. (2) Computation of the difference between the image and the Σ-Δ mean (motion likelihood measure). (3) Computation of the Σ-Δ variance defined as the Σ-Δ mean of N times the non-zero differences. (4) Computation of the motion label by comparison between the difference and the variance.

Figures 2 to 4 display an example of the evolution over time of the different values computed as above, for three pixels extracted from a country scene similar to the image shown on Figure 1, for a 1000 frames sequence. The red solid line represents the input image I_t. The blue dashed line corresponds to the Σ-Δ mean M_t. The green dashed line represents the difference Δ_t. Finally, the purple dotted line is the Σ-Δ variance V_t (using $N = 4$). The detection field D_t is not represented explicitly, but corresponds to the

Boolean indicator of the condition “the green dashed line is over the purple dotted one”.

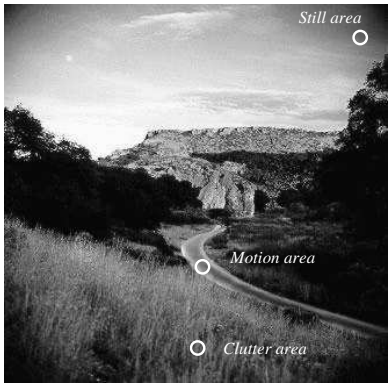


Figure 1: Example of observed scene, with 3 particular pixels from 3 different areas.

The pixel used in Figure 2 is a pixel in a still zone, with flat temporal activity, such as a remote area of the static background (in this example, a sky lightly covered with slowly moving clouds). For such pixels, the high frequency variation corresponds to temporal noise due to the acquisition and digitization processes. The low frequency variations are due to illumination changes or slow motion of low contrast objects.

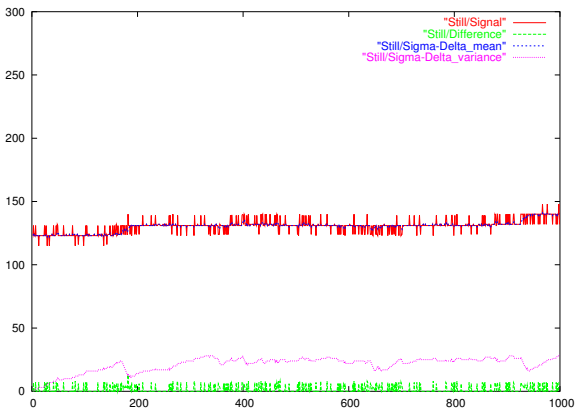


Figure 2: Temporal variation of a pixel value in a still area.

The pixel used in Figure 3 is a pixel in a motion area, such as tracks or corridors (in this example, a country road with 4 vehicles passing away). In that case, the moving objects give rise to sharp changes that are not taken into account by the Σ-Δ mean, and then the difference field shows a peak. Such peaks are discriminated thanks to the comparison with the Σ-Δ variance.

The pixel used in Figure 4 is a pixel in a clutter area, i.e. a zone of physical changes due to intrinsic nature of the

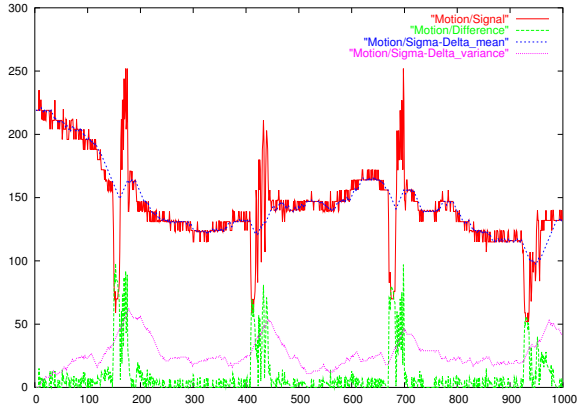


Figure 3: Temporal variation of a pixel value in a motion area.

scene rather than moving objects. Examples of such areas are: trees moving with the wind, river, or crowd in a urban scene (in our example, high grass in the foreground of the scene). In that case, the difference field shows a repetition of peaks, and if these peaks are close enough from each other with respect to the delay induced by Σ-Δ modulation, then they will be taken into account in the Σ-Δ variance, in such a way that the difference will remain less that the variance.

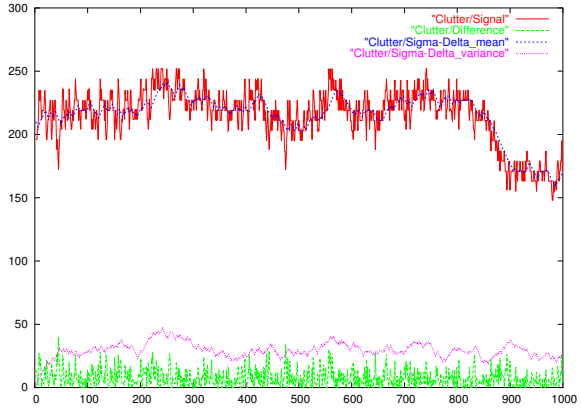


Figure 4: Temporal variation of a pixel value in a clutter area.

Figure 6 (1)-(4) displays the result of this method for one frame of an urban traffic sequence. The four images represent respectively I_t, M_t, V_t and D_t. It can be seen that the discrimination of “moving” pixels corresponds to the detection of temporally *salient* pixels with respect to the temporal activity. This allows to discard irrelevant (clutter) motion, but also to be less sensitive to sensor oscillations, as it is shown on Figure 6 (5) and (6): A uniform random oscilla-

tion of ± 1 pixels has been simulated on the same sequence. In this case, M_t converges to an approximation of the spatiotemporal median, and then V_t (Fig. 6(5)) emphasizes the regions of high contrast, thus increasing the local threshold in these regions, and avoiding the detection of the whole scene contour in D_t (Fig. 6(6)).

The Σ-Δ background estimation provides a simple and efficient method to detect the significantly changing pixels in a static scene, with respect to a time constant depending on the number of gray levels of the images. Nevertheless it is a pure temporal processing, which can only provide pixel-level detection. In the following section, we use a Markovian framework to improve the motion detection by exploiting the spatial correlation in the data.

3. Spatial regularization

We follow the model that has been used for real-time implementation in different architectures in [1] and in [7]. This Markov model is based on the estimation of a binary (background/foreground) motion field e given an *observation field* o, by maximizing a Bayesian *maximum a posteriori* criterion, i.e. given a realization of the observation field $o = y$, finding the realization x of the motion label field e that maximizes the conditional probability $P(e = x/o = y)$. Under the hypothesis that e is a Markov field, and a probabilistic model linking o and e, this corresponds to finding the motion field e that minimizes the global *energy* function defined over the set of pixels $\mathbb{S}$ as follows:

$$U = \sum_{s \in \mathbb{S}} [U_m(e(s)) + U_a(e(s), o(s))],$$
$$\text{with } U_m(e(s)) = \sum_{r \in \mathcal{V}(s)} V_e(e(s), e(r)),$$
$$\text{and } U_a(e(s), o(s)) = \tfrac{1}{2\sigma^2}[o(s) - \Psi(e(s))]^2.$$

$U_m(e(s))$ is called *model energy* and is designed to provide spatiotemporal regularity in the motion field. It is based on the Markovian modeling of e as a Gibbs field, where $\mathcal{V}$ is the set of neighbors of the pixel s, and the potential functions $V_e(e(s), e(r))$ equals $-\beta_{sr}$ if $e(s) = e(r)$, and $+\beta_{sr}$ if $e(s) \neq e(r)$. The β_{sr} are positive constants, whose values depend on the nature of the neighborhood. We use a uniform 6-connected spatiotemporal topology with 3 different values $\beta_S = 20$ for the 4 spatial neighbors, $\beta_P = 10$ for the past neighbor, and $\beta_F = 30$ for the future neighbor.

$U_a(e(s), o(s))$ is called *fitness energy* and is designed to ensure a certain level of attachment to the input data, i.e. the observation o. This term comes from the conditional probability of the observation field o, with respect to the motion field e, assuming that $o(s) = \Psi(e(s)) + n(0, \sigma^2)$, with $n(0, \sigma^2)$ a centered Gaussian noise of variance σ^2, $\Psi(e(s)) = 0$ if $e(s)$ has the background value, and $\Psi(e(s)) = \alpha$ if $e(s)$ has the foreground value. [1] and [7] use the absolute difference between two consecutive frames as the observation field. They use a constant value for α (20), and estimate σ^2 by computing the spatial variance of the observation.

The minimization of the global energy U is realized by the deterministic relaxation called iterated conditional mode (ICM): all the pixels are sequentially updated, and each pixel s is given the label $e(s)$ corresponding to the smallest local energy $U_m(e(s)) + U_a(e(s), o(s))$. Usually, instead of a true relaxation, a limited number of scans is performed (typically 4). This algorithms is known to be very sensitive to the quality of the initial value of the estimated motion field. [1] and [7] use a threshold of the observation (i.e. the absolute difference between two consecutive frames) as the initial estimation of e.

In our algorithm, we use the same model, with the following exceptions (referring to the variables used in Table 1):

- for the observation o, we use Δ the difference between the background and the current frame.
- we use the Σ-Δ variance V as a second observation field, to estimate locally the dispersion factor: $(\frac{V}{N})^2$ is used instead of σ^2 for weighting the relative importance of U_a with respect to U_m.
- the initialization of the Markovian relaxation corresponds to the pixel-level detection D.

What are the advantages of this algorithm compared to the original Markovian model ?

First, the difference with the Σ-Δ background is more robust than the frame to frame difference, because it combines information over a large period of time instead of two frames. It is much less sensitive to the aperture problem, which makes difficult the detection of large homogeneous zones in motion. It is also less dependent on the velocity of the objects.

Next, for the same reasons, D is in general much better to initialize the relaxation than a binarized frame difference, because it is closer to the expected solution. It must be emphasized that, for the ICM algorithm, once the initialization computed, the other parameters of the model are not critical, and have shown good behaviors on lots of different sequences.

Finally, the dispersion parameter is computed locally, then no global computation is needed at each frame. This allows the computation of the whole algorithm using only local memory sharing, thus permitting a massive spatial parallelism.

To increase the confidence in the detected objects, a higher level of processing, is needed, involving regional and

global computation. This is discussed in the following section.

4. Region level processing

A precise description of the region level processing is not in the scope of this paper. We will only present its principles and discuss the feedback that can be done on the local computation in order to increase the robustness of the background estimation.

The output of the Markovian regularization is a low-level estimation of the foreground, consisting of the temporal salient pixels presenting spatial correlation. In order to enhance the quality of detection and lower the false alarm rate, some higher level processing is needed, using regional and global computations.

The pixels are grouped into regions representing objects. It is usually done by connected components labeling followed by fusion. The resulting objects then undergo morphological filtering, which can reject some objects under size or shape criteria. Kinematic filtering can also be employed in order to discriminate the objects whose motion is consistent with regard to the application (e.g. car, pedestrian,...).

In addition to the diminution of the false alarm rate, the interest of the global level processing is to allow a frame rate feedback on the low-level detection. One of the most straightforward example is the adaptation to a sudden change of background: if a global confidence index in the background (e.g. the relative surface area occupied by the foreground) decreases beneath a certain level, the decision can be made to re-initialize the background, in order to lower the re-adaptation time.

Another useful feedback is to attach a confidence index c to each filtered object, from 1 (lowest confidence) to ∞ (absolute confidence). Those indices are then used as a period of update for the Σ-Δ estimation: if the pixel s belongs to the foreground with confidence $c(s)$, then $M_t(s)$ and $V_t(s)$ are updated only every $c(s)$ frames. This enhances the quality of detection by increasing the robustness of the non-linear filter, and avoiding the objects stopping temporarily (such as cars at red light) to enter in the background too quickly (thus generating "ghost" when they move again).

Figures 6(9) and (10) show an application of the relevance feedback, using a uniform confidence index of 6 over a mask which is simply a morphological filtering of the spatial regularization output.

5. Conclusions

We have presented a new algorithm allowing a robust and accurate detection of moving objects for a small cost in memory consumption and computational complexity. We have emphasized the nice properties of the Σ-Δ filter for the detection of salient features in time-varying signal, showing that the interest of such filter goes well beyond its temporal median convergence property.

We have adapted a classical Markovian model to perform the spatial regularization that is needed to eliminate irrelevant salient pixels and aggregate the relevant ones in significant region. The multiple observation field produced by the Σ-Δ estimation has shown relevant as input of the Markovian relaxation algorithm.

Because it only relies on pixel-wise or spatially limited interactions, the whole low-level processing (temporal processing and spatial regularization) is suited to a massively parallel implementation. We are at the present time implementing the algorithm on a programmable artificial retina [6], which is a fine-grained parallel machine with optical input. The algorithm is indeed well adapted to the architecture, which consists in a mesh of tiny processors with limited memory and computation power. We have already implemented an alternate version of the algorithm (same temporal processing followed by a spatiotemporal morphological filtering) with the following performances, for a 200x200 retina array running at 25 Mhz, using 8 bits per pixels: 2.25 ms per frame, of which only 0.75 ms for the sole computation, and the rest for the acquisition.

The present limitation of our approach lies in the adaptation capability to certain complex scenes. It is the case for very low motion, that is likely to be taken as a background characteristics. A straightforward way to address this problem is to downsample the updating rate of the estimates, i.e. to use only every nth frame to compute M_t and every pth frame to compute V_t. See Figure 5 for an example of result (in this sequence, the scene is always full with persons moving then stopping for a while, and the camera is slightly oscillating). But this adds two learning rate parameters that can be critical, because reducing the updating rate increases the sensitivity to oscillations and the adaptation to a new scene. Another limitation case is the wide amplitude periodical motion (e.g sea surge), that will be classified as foreground if the period is too long. We are investigating the possibilities to go beyond these limits by combining different variance models using different sampling periods in the Σ-Δ estimation, in order to get a richer quantitative estimation of the motion activity.

In future works, we will also focus on more sophisticated high-level filtering, in order to increase the robustness of the background estimation, and then of the whole detection.

References

[1] A. Caplier, C. Dumontier, F. Luthon, and P. Coulon. Mrf based motion detection algorithm image processing board implementation. *Traitement du signal (in french)*, 1996.

(1) (2)

Figure 5: Result on a complex sequence, using downsampling parameters ($n = 10$ and $p = 4$). (1) Detection result (contour superimposed on the original image) (2) Σ-Δ mean (used by courtesy of L. Lacassagne UPS/IEF)

[2] T. H. Chalidabhongse, K. Kim, D. Harwood, and L. Davis. A perturbation method for evaluating background subtraction algorithms. In *Proc. Joint IEEE Int. Workshop on Visual Surveillance and Performance Evaluation of Tracking and Surveillance*, Nice - France, 2003.

[3] S.-C. Cheung and C. Kamath. Robust techniques for background subtraction in urban traffic video. In *Proc. SPIE Video Communications and Image Processing*, San Jose - CA, 2004.

[4] A. Elgammal, D. Harwood, and L. Davis. Non-parametric Model for Background Subtraction. In *Proc. IEEE European Conference on Computer Vision*, Dublin - Ireland, 2000.

[5] K.-P. Karmann and A. von Brandt. *Time-Varying Image Processing and Moving Object Recognition*, chapter Moving Object Recognition Using an Adaptive Background Memory. Elsevier, 1990.

[6] T. Komuro, I. Ishii, M. Ishikawa, and A. Yoshida. A digital vision chip specialized for high-speed target tracking. *IEEE Trans. on Electron Devices*, 2003.

[7] L. Lacassagne, M. Milgram, and P. Garda. Motion detection, labeling, data association and tracking in real-time on risc computer. In *Proc. IEEE ICIAP*, pages 520–525, 1999.

[8] B. Lee and M. Hedley. Background estimation for video surveillance. In *IVCNZ02*, pages 315–320, 2002.

[9] N. McFarlane and C. Schofield. Segmentation and tracking of piglets in images. *Machine Vision and Applications*, 8:187–193, 1995.

[10] A. Mittal and N. Paragios. Motion-based background subtraction using adaptive kernel density estimation. In *Proc. IEEE CVPR*, 2004.

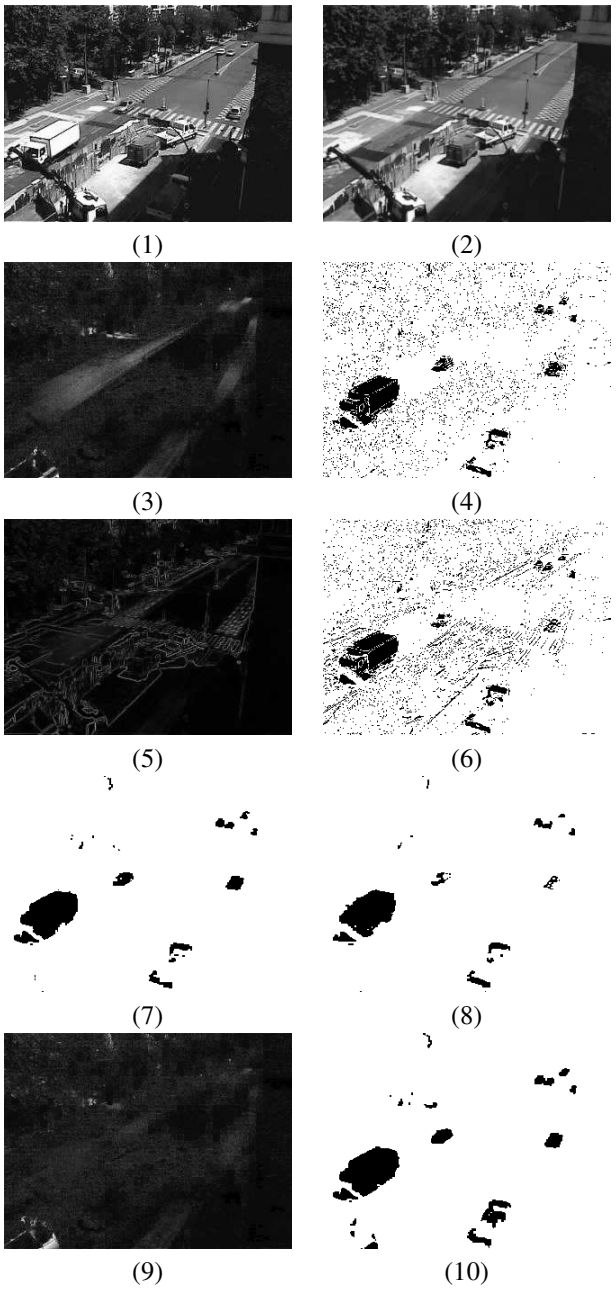
[11] N. Oliver, B. Rosario, and A. Pentland. A bayesian computer vision system for modeling human interactions. *IEEE Trans. on PAMI*, 2000.

[12] M. Piccardi. Background subtraction techniques: a review. http://www-staff.it.uts.edu.au/∼massimo/.

[13] C. Stauffer and E. Grimson. Learning patterns of activity using real-time tracking. *IEEE Trans. on PAMI*, 2000.

[14] K. Toyoma, J. Krumm, B. Brumitt, and B. Meyers. Wallflower: Principles and Practice of Background Maintenance. In *Proc. IEEE ICCV*, pages 255–261, Kerkyra - Greece, 1999.

[15] C. Wren, A. Azarbayejani, T. Darrell, and A. Pentland. Pfinder: Real-time tracking of the human body. *IEEE Trans. on PAMI*, 1997.



(1) (2) (3) (4) (5) (6) (7) (8) (9) (10)

Figure 6: Result of the proposed algorithm on a traffic sequence. (1) I_t (2) M_t. (3) V_t (displayed with normalized histogram). (4) D_t (N=2). (5) V_t with simulated oscillations of the camera (Normalized histogram). (6) D_t for the oscillating camera (N=2). (7) Detection after Markovian regularization (5 iterations). (8) *idem* for the oscillating camera. (9) V_t using relevance feedback (applying the same transformation as image (3). (10) Detection using relevance feedback on the pixel level processing and Markovian regularization.