

Multi-frame motion detection for active/unstable cameras

Mirko Ristivojević and Janusz Konrad

Department of Electrical and Computer Engineering, Boston University

8 Saint Mary's St., Boston, MA 02215

[mirko, jkonrad]@bu.edu

Abstract—Network cameras, extensively used in video surveillance, often allow pan-tilt-zoom functionality and are also subject to wind load and mount vibrations, thus causing video frame misalignment. Although algorithms for motion detection, a basic block of most visual surveillance systems, are relatively mature for fixed cameras, they usually perform poorly for active and/or vibrating cameras. The issue is particularly severe for algorithms using multiple video frames jointly. In this paper, we extend our earlier work on multiple-frame motion detection to the case of active and unstable cameras. Our method accounts for spatially-affine, inter-frame transformations that can vary in time, uses a variational formulation and applies a level-set solution. We present ground-truth and real-data experimental results and show significant improvements over earlier methods.

Keywords—motion detection, multi-frame video analysis, level sets, affine motion

I. INTRODUCTION

Network camera has become a sensing device of choice for airport, urban, and highway surveillance. The dramatic reduction of its cost and size in recent years is likely to continue in the near future thus leading to a further proliferation of network cameras but also to the emergence of new challenges in their use [1], [2], [3], [4]. One challenge that is already apparent is related to camera movements.

Many network cameras are mounted outdoors and are subject to wind load and/or mount vibrations, thus causing camera sway and jitter. Furthermore, some cameras include pan-tilt-zoom (PTZ) functionality. In either case, the captured video frames are misaligned with respect to one another due to camera movements. While algorithms for motion detection, a key tool in visual surveillance, are relatively mature for fixed cameras (e.g., background subtraction [5], [6], [7], active contours [8], [9], [10], Markov models [11], [12], [13]), they do not perform accurately for active or vibrating cameras. The issue is particularly serious for motion detection methods using multiple frames jointly; a single misaligned pixel in one frame may affect the detection result in several neighboring video frames. However, such methods are of great interest since it has been

shown that they outperform equivalent two-frame methods (see [14], [10] for examples). One should note at this point that although background subtraction methods often use many past frames, this is done recursively rather than jointly, i.e., while past frames impact the current detection result, a current frame does not affect past detection results.

Although, in principle, camera motion can be estimated first and then image sequence can be compensated for camera motion prior to object motion detection, such a two-step approach does not permit mutual interaction between compensation and detection steps; once a compensation error is committed, no recovery is possible. An alternative is to embed camera motion compensation into the motion detection algorithm. We adopt the latter approach here and extend our recent multi-frame, level-set motion detection method [10]. In particular, we incorporate a spatially-affine transformation applied to consecutive video frames in order to account for the misaligned background. The affine model is accurate under orthographic projection (i.e., for far away backgrounds) for camera zoom, pan, tilt and roll (in addition to track and boom) [15]. Furthermore, we allow the affine parameters to change in time thus permitting time-varying zoom and motion of the camera. We present experimental results on ground-truth and real data confirming usefulness of the proposed approach. In particular, we demonstrate a significant reduction of the detection error vis-à-vis recent methods using other frame alignment models [16].

This paper is organized as follows. In Section II, we present the case of motion detection for static cameras. In Section III, we extend the static-camera approach by incorporating a time-varying affine motion model. We show experimental results in Section IV, and draw conclusions in Section V.

II. MULTI-FRAME MOTION DETECTION FOR STATIC CAMERAS

Let $I(\mathbf{x}, t)$, where \mathbf{x} is spatial position and t is time, be a continuous dynamic image, and let $\Omega \times \mathcal{T}$ be its spatio-temporal domain, i.e., $\mathbf{x} \in \Omega, t \in \mathcal{T}$. Also, let ζ be a parametrized surface defined on $\Omega \times \mathcal{T}$ that delineates a moving object through time (often referred to as object tunnel or object tube). Since the camera is static, we assume that intensity of the background remains approximately constant over time, i.e., $I(\mathbf{x} = \mathbf{x}_0, t) \approx \text{const}$. We also

M. Ristivojević was with Boston University when this research was performed. He is currently with Intellivid Corp., Cambridge MA 02138, USA.

This work was supported in part by the National Science Foundation under grants CCR-0209055 and INT-0233318.

assume that the surface $\vec{\zeta}$ is smooth, for example that its area \mathcal{S} is small.

Numerous variational formulations have been proposed to date for motion detection and segmentation from two video frames [17], [18], [19]. As for variational formulations of multi-frame motion detection, there are two types: boundary-based and volume-based. Boundary-based formulations [20], [16] rely on a motion characteristic, such as the normal velocity component, being sufficiently distinct at a moving object boundary. Clearly, the positive vote counts only at the boundary that usually occupies a small fraction of the domain. Volume-based formulations [21], [22], [9] rely on a motion characteristic, often the temporal intensity gradient, being sufficiently different between moving object and (static) background. Since the positive vote counts for all pixels inside the object, not only on its boundary, volume-based approaches tend to be more reliable. This is the approach we will pursue here.

Let $\xi(\mathbf{x}, t)$ and $\bar{\xi}(\mathbf{x}, t)$ be measures of temporal activity in a dynamic image at space-time location (\mathbf{x}, t) inside and outside of $\vec{\zeta}$, respectively. Then, the surface $\vec{\zeta}$ can be computed *via* volume competition, i.e., by minimizing a functional balancing cumulative activity measure (inside and outside of $\vec{\zeta}$), and surface roughness expressed as follows:

$$\begin{aligned} \min_{\vec{\zeta}} & \iint_{\mathcal{V}} \xi(\mathbf{x}, t) d\mathbf{x} dt + \\ & \iint_{\bar{\mathcal{V}}} \bar{\xi}(\mathbf{x}, t) d\mathbf{x} dt + \\ & \lambda \iint_{\mathcal{S}} d\vec{\zeta}, \end{aligned} \quad (1)$$

where $\vec{\zeta} = \partial\mathcal{V}$, \mathcal{V} is the inside of $\vec{\zeta}$, $\bar{\mathcal{V}}$ is the outside of $\vec{\zeta}$, and λ associates a cost with the Euclidean area element $d\vec{\zeta}$. The first term above measures temporal activity inside the surface $\vec{\zeta}$ (object tunnel), whereas the second term – outside of $\vec{\zeta}$. The third term assures a minimum-area (smooth) surface $\vec{\zeta}$. Thus, the minimization above seeks as smooth a surface as possible that divides the domain $\Omega \times \mathcal{T}$ into such \mathcal{V} and $\bar{\mathcal{V}}$ that they are most compatible with ξ and $\bar{\xi}$, respectively.

A simple, yet efficient, model uses a fixed cost α within the moving object, and absolute frame difference in the background [21]:

$$\begin{aligned} \xi(\mathbf{x}, t) &= \alpha, \\ \bar{\xi}(\mathbf{x}, t) &= |I(\mathbf{x}, t) - I(\mathbf{x}, t - 1)|. \end{aligned} \quad (2)$$

In order that a point be labeled as background (i.e., not moving or in $\bar{\mathcal{V}}$), its temporal intensity variation $\bar{\xi}$ must be small as otherwise it would be more optimal to assign it to the object due to its fixed cost α . On the other hand, in order that this point be labeled as object (i.e., in \mathcal{V}), its intensity variation must be large as otherwise it would be

more optimal to assign it to the background. The balance between such assignments is controlled by the parameter α .

Considering the above model, the minimization (1) reduces to:

$$\min_{\vec{\zeta}} \iint_{\Omega \times \mathcal{T}} h(I) d\mathbf{x} dt + \lambda \iint_{\mathcal{S}} d\vec{\zeta}, \quad (3)$$

where

$$h(I) = \begin{cases} \alpha & \text{if } (\mathbf{x}, t) \in \mathcal{V}, \\ |I(\mathbf{x}, t) - I(\mathbf{x}, t - 1)| & \text{if } (\mathbf{x}, t) \in \bar{\mathcal{V}}. \end{cases}$$

and leads to the following evolution equation [9]:

$$\frac{\partial \vec{\zeta}}{\partial \tau} = [\alpha - |I(\mathbf{x}, t) - I(\mathbf{x}, t - 1)| + \lambda \kappa_m] \vec{n}, \quad (4)$$

where τ is the *evolution* time (the time in the video sequence is denoted by t), κ_m is the curvature of surface $\vec{\zeta}$, and \vec{n} is the inward unit normal vector of $\vec{\zeta}$.

Implementation of evolution (4) using level-set methodology has become popular in the last decade for its flexibility (topology independence) and stability. Employing this methodology, evolution (4) can be shown to be equivalent to the following evolution of a (higher-dimensional) level-set surface ϕ :

$$\frac{\partial \phi}{\partial \tau} = [\alpha - |I(\mathbf{x}, t) - I(\mathbf{x}, t - 1)| + \lambda \kappa_m] \|\nabla \phi\|.$$

This equation can be implemented using standard discretization as described in [23]. In a full implementation, one usually calculates the evolution force at zero level-set points, extends this force, e.g., using the fast marching algorithm, updates the surface ϕ , and re-initializes it. Alternatively, one can use a recently-proposed fast implementation [24], [25] although at a slight loss of precision.

Clearly, the above approach performs well on sequences void of camera motion, however when camera moves either intentionally (pan-tilt-zoom) or unintentionally (vibrations, wind load) the background between frames becomes misaligned thus causing errors.

III. MULTI-FRAME MOTION DETECTION FOR ACTIVE/UNSTABLE CAMERAS

In order to account for camera motion in the volume-competition formulation (1), the temporal activity measures ξ and $\bar{\xi}$ (2) need to be modified.

We propose to use a spatially-affine motion model that is accurate under orthographic projection (i.e., for far away backgrounds) for camera zoom, pan, tilt and roll [15], and, therefore, especially useful in outdoor scenes under camera vibration and operator manipulation. Furthermore, we permit variation of affine parameters between consecutive video frames in order to account for the dynamics in camera zoom or motion.

While keeping a fixed cost within the moving object, we modify the temporal activity measure in the background $\bar{\xi}$ (2) to account for the camera motion as follows:

$$\begin{aligned}\xi(\mathbf{x}, t) &= \alpha, \\ \bar{\xi}(\mathbf{x}, t) &= \rho(I(\mathbf{x}, t) - I(\mathbf{x} - \mathbf{d}_{\bar{\theta}_t}(\mathbf{x}, t), t - 1)),\end{aligned}\quad (5)$$

where $\mathbf{d}_{\bar{\theta}_t}(\mathbf{x}, t)$ is a displacement vector calculated from global motion parameters $\bar{\theta}_t$ that are time-varying (subscript t). Clearly, $\mathbf{d}_{\bar{\theta}_t}$ describes background motion, resulting from camera dynamics, between frames at times $t - 1$ and t . For motion parameters $\bar{\theta}_t = (p_t^1 \ p_t^2 \dots p_t^6)^T$, we use a dynamic affine model:

$$\mathbf{d}_{\bar{\theta}_t}(\mathbf{x}, t) = \begin{bmatrix} p_t^1 \\ p_t^2 \end{bmatrix} + \begin{bmatrix} p_t^3 & p_t^4 \\ p_t^5 & p_t^6 \end{bmatrix} (\mathbf{x} - \mathbf{x}_0), \quad (6)$$

where \mathbf{x}_0 is a reference spatial coordinate (fixed or dependent on previously-detected background shape, e.g., centroid). Note, that in (5) we use a robust M-estimator ρ (e.g., absolute value, Lorentzian, Geman-McClure function) instead of a quadratic in order to improve method's robustness to outliers [26], [27]. We do so since during iterative computation of an intermediate segmentation surface its shape may be different from the true, underlying object boundary, and thus some points inside $\bar{\mathcal{V}}$ may not belong to the background and cannot be explained by the background motion model. Using a robust error measure helps reduce the impact of such inconsistencies (outliers) on the estimation process.

With the above definition of $\bar{\xi}$, the minimization (1) becomes:

$$\begin{aligned}\min_{\{\bar{\zeta}, \bar{\theta}_t\}} & \iiint_{\bar{\mathcal{V}}} \alpha d\mathbf{x}dt + \\ & \iiint_{\bar{\mathcal{V}}} \rho(I(\mathbf{x}, t) - I(\mathbf{x} - \mathbf{d}_{\bar{\theta}_t}(\mathbf{x}, t), t - 1)) d\mathbf{x}dt + \\ & \lambda \iint_{\bar{\mathcal{S}}} d\bar{\zeta}.\end{aligned}\quad (7)$$

A minimum is reached in (7) when the surface $\bar{\zeta}$ partitions the domain $\Omega \times \mathcal{T}$ in such a way that points (\mathbf{x}, t) with small motion-compensated intensity difference are assigned to the outside of $\bar{\zeta}$ (i.e., $\bar{\mathcal{V}}$), and those with large difference are assigned to the inside of $\bar{\zeta}$ (i.e., $\bar{\mathcal{V}}$). In other words, image points that can be affine-compensated using camera motion are assigned to the background, whereas those that cannot are assigned to the object.

We solve minimization (7) by decomposing it into two interleaved minimizations: estimation of motion parameters $\bar{\theta}_t$ given a segmentation surface, and estimation of segmentation surface $\bar{\zeta}$ given motion parameters.

Assuming the segmentation surface is known ($\bar{\zeta}^*$ and $\bar{\mathcal{V}}^*$),

minimization (7) with respect to $\bar{\theta}_t$ reduces to:

$$\min_{\bar{\theta}_t} \iiint_{\bar{\mathcal{V}}^*} \rho(I(\mathbf{x}, t) - I(\mathbf{x} - \mathbf{d}_{\bar{\theta}_t}(\mathbf{x}, t), t - 1)) d\mathbf{x}dt, \quad (8)$$

because the remaining integrals in (7) are independent of $\bar{\theta}_t$. Considering motion parameters $\bar{\theta}_t$ at different time instants t to be independent and discretizing the formulation (8), leads to the following minimization for each $\bar{\theta}_{t_k}$:

$$\min_{\bar{\theta}_{t_k}} \sum_{\mathbf{x}_i \in \bar{\mathcal{R}}_{t_k}^*} \rho(I[\mathbf{x}_i, t_k] - I[\mathbf{x}_i - \mathbf{d}_{\bar{\theta}_{t_k}}[\mathbf{x}_i, t_k], t_{k-1}]), \quad (9)$$

where $\bar{\mathcal{R}}_{t_k}^*$ is the background region in frame at discretized time t_k , easily computed as the cross-section of volume $\bar{\mathcal{V}}^*$ at time t_k . We solve the above minimization using the Broyden-Fletcher-Goldfarb-Shanno quasi-Newton method (`fminunc` function in *Matlab*).

Now, that the new motion parameters $\bar{\theta}^*$ have been computed, we solve minimization (7) for $\bar{\zeta}$:

$$\begin{aligned}\frac{\partial \bar{\zeta}}{\partial \tau} &= [\alpha - \rho(I(\mathbf{x}, t) - I(\mathbf{x} - \mathbf{d}_{\bar{\theta}_t}(\mathbf{x}, t), t - 1)) \\ &+ \lambda \kappa_m] \bar{\mathbf{n}},\end{aligned}\quad (10)$$

using the level-set methodology as follows:

$$\begin{aligned}\frac{\partial \phi}{\partial \tau} &= [\alpha - \rho(I(\mathbf{x}, t) - I(\mathbf{x} - \mathbf{d}_{\bar{\theta}_t}(\mathbf{x}, t), t - 1)) \\ &+ \lambda \kappa_m] \|\nabla \phi\|.\end{aligned}\quad (11)$$

Note that since between consecutive iterations of surface evolution only small changes in $\bar{\zeta}$ take place, changes in the corresponding motion parameters are small as well. Therefore, we estimate parameters $\bar{\theta}_{t_k}$ for each frame pair by solving (9) only every few iterations of surface evolution (11), i.e., not until the change in $\bar{\zeta}$ becomes significant. This procedure is repeated until no further improvement in the estimated motion and partitioning surface can be obtained.

However, which minimization, (9) or (11), should be performed first? Usually it is easier to compute a reasonable estimate of global frame motion $\bar{\theta}_t$ than that of the surface $\bar{\zeta}$ (object tunnel). Thus, the first step of the overall algorithm is an initial estimation of global motion with no information about object location; the minimization (9) must thus be performed on the whole frame I_{t_k} as $\bar{\mathcal{R}}_{t_k}$ is unknown. In order to avoid local minima in this case and handle large background motion, this minimization should not be initialized with translation parameters (p_t^1, p_t^2) of the affine model (6) set to zero. For example, these parameters should be either obtained by phase correlation or by block matching applied to the whole frame [15]. With such initial translations approximated, minimization (9) has a good chance of finding correct parameters $(p_t^3, p_t^4, p_t^5, p_t^6)$.

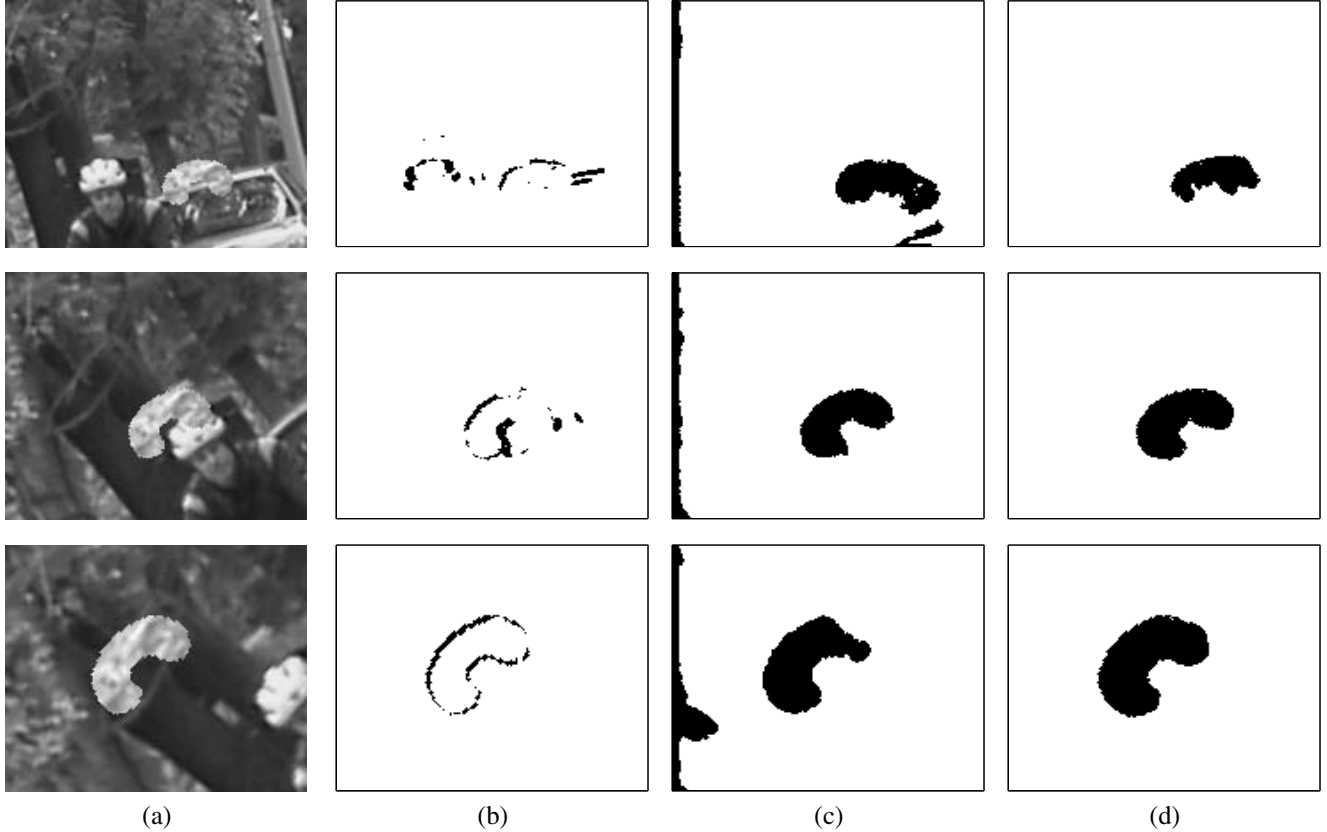


Figure 1. (a) Three frames from natural-texture, synthetic-motion sequence *Bean*; and motion detection results for: (b) no camera motion model ($\alpha = 100, \lambda = 1$); (c) spatially- and temporally-constant translation ($\alpha = 0.9, \lambda = 0.1$); and (d) spatially-affine, time-varying model ($\alpha = 0.8, \lambda = 0.1$). See text for details.

IV. EXPERIMENTAL RESULTS

We have evaluated the performance of our approach on both ground-truth and real-data imagery using a fast level-set implementation [24], [25] that affords a significant speedup over standard implementations.

In order to measure gains offered by the proposed spatially-affine, time-varying camera motion model we have compared it directly with two approaches:

- 1) variational, multi-frame formulation that ignores camera motion [10],
- 2) variational, multi-frame formulation that uses spatially-translational, temporally-constant camera motion model [16].

We implemented all three approaches *via* minimization (7) with temporal activity measures (5) adapted as follows:

- approach ignoring camera motion [10]: $\vec{d}_{\vec{\theta}_t} = \vec{0}$, absolute-value estimator $\rho(x) = |x|$ simplifying (5) to (2),
- translational model (parameters p^1 and p^2 are independent of video time t) [16]: $\vec{d}_{\vec{\theta}_t}$ derived from $\vec{\theta}_t = (p^1 \ p^2)^T$, Geman-McClure M-estimator $\rho(x) = x^2/(1+x^2)$,

- affine model proposed here: $\vec{d}_{\vec{\theta}_t}$ derived from $\vec{\theta}_t = (p_t^1 \ p_t^2 \dots p_t^6)^T$, Geman-McClure M-estimator $\rho(x) = x^2/(1+x^2)$.

Note that while we use the spatially-translational, temporally-constant camera motion model proposed in [16], we minimize a different cost functional than originally proposed by the authors in order to avoid mixing the impact of the model with that of the cost functional and optimization method. Also, note that since the approach ignoring camera motion uses different type of estimator (absolute value), the resulting range of ξ values is also different and thus the regularization coefficient λ balancing surface smoothness and segmentation accuracy must be adjusted. We used $\lambda = 1$ for the approach ignoring camera motion (absolute value estimator) and $\lambda = 0.1$ for the other two approaches (Geman-McClure estimator).

First, we compared the three approaches on a natural-texture, synthetic-motion sequence (ground truth sequence) in which both a bean-shaped object and background undergo accelerated zoom-in and rotation, each with different parameters (Fig. 1). From among many results for different values of parameter α , we show those with the lowest segmentation error (symmetric difference between ground truth and actual

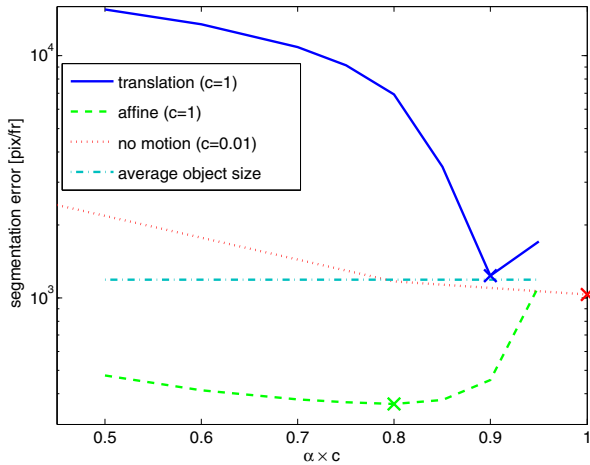


Figure 2. Average motion detection error [pixels/frame] for experiments with different values of parameter α , on sequence from Fig. 1. The cross on each plot (\times) marks the error and α for results from Fig. 1. The horizontal “dash-dot” line shows average object size. The scaling factor c is used only to overlay the three plots on the same graph.

segmentation) in each case. Note a poor segmentation accuracy for the algorithm unaware of camera motion (Fig. 1.b); the lowest error, at $\alpha = 100$, corresponds to the detection of object boundaries only (for $\alpha = 20$ errors occur in the object and throughout the background). The result for spatially- and temporally-constant translational camera motion model, shown in Fig. 1.c, is far more accurate, but the moving object still has significant errors and one image boundary is interpreted as having the same motion as the object (due to translational camera model used on rotating and zooming-in background). The proposed method (Fig. 1.d) results in a more precise object mask than the translational model and accurately handles the image boundary.

The average segmentation error per pixel for each sequence is shown in Fig. 2 as a function of parameter α (3). The different range of α for the model unaware of camera motion (α close to 100) as opposed to the other two models (α close to 1) is due to the use of absolute value instead of a robust measure ρ in $\bar{\xi}$. Note that the scaling factor c in Fig. 2 is used solely for visualization purposes (so that the three plots coincide). Clearly, the method based on the time-varying affine model outperforms the one based on translation by a wide margin for $\alpha < 0.9$ (note a log scale on the error axis). For $\alpha \geq 0.95$ the object cannot be detected, no matter what motion model, so the segmentation error is close to the size of the object (horizontal line). The object is very accurately detected if affine model is used with α between 0.65 and 0.85. Note that although the method ignoring camera motion outperforms numerically the translational model, it either cannot detect the object (Fig. 1.b) or produces a noisy segmentation in the background (lower α).

Since the approach ignoring camera motion, as expected,

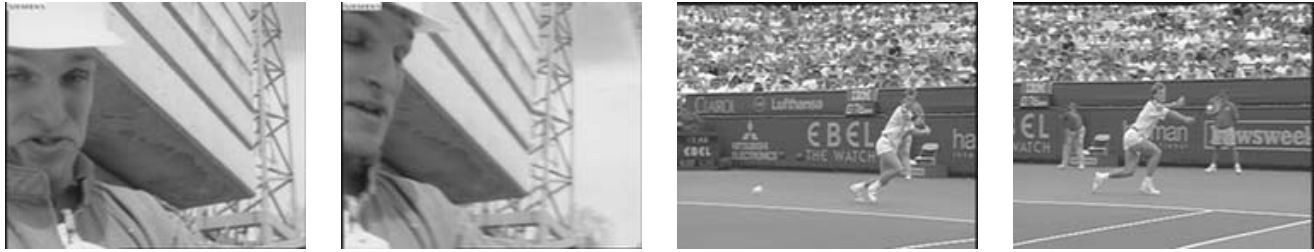
performs poorly when the background moves, a comparison on natural sequences with significant camera pan was performed on the other two approaches only (Fig. 3). While the translational model produces significant errors in the (moving) background, the affine model produces quite accurate segmentations. The upper body in *Foreman* is correctly detected although some motion boundaries are not accurate due to lack of texture, e.g., on the helmet (no matter what displacement, similar intensity error occurs). Note, however, that the global frame partitioning into moving object (upper torso) and differently-moving background is overall correct. In *Stefan*, the body of a tennis player is very accurately outlined, unlike in the result obtained by the constant-translation method where the body is very hard to discern; most detections in this case occur in the background (spectators) since the model cannot accurately handle camera pan.

V. CONCLUSIONS

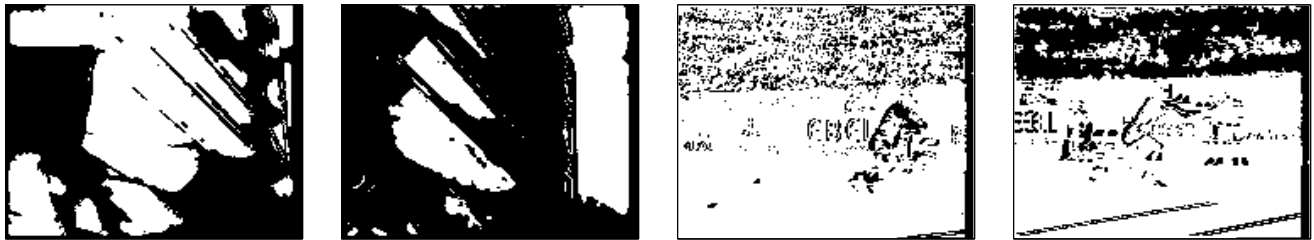
Motion detection, in case of a non-static camera, requires global motion compensation as a pre-processing step. Such a two-step approach, however, does not permit interaction between global motion compensation and motion detection. We have proposed a variational formulation that incorporates global motion compensation based on a spatially-affine, time-varying model. Experimental results have confirmed expected improvements in motion detection quality in the presence of camera motion. An issue to be addressed in future work is the lag of moving object boundaries behind the actual object. It is caused by newly-exposed pixels that cannot be explained by global motion and thus are included in the object. A correct way of solving this problem is to model and estimate occlusion and newly-exposed areas explicitly.

REFERENCES

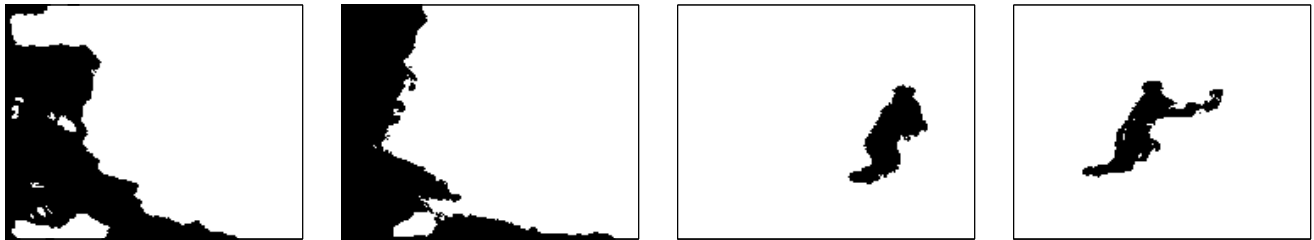
- [1] G. L. Foresti, C. Micheloni, L. Snidaro, P. Remagnino, and T. Ellis, “Active video-based surveillance system,” *IEEE Signal Process. Mag.*, vol. 22, no. 2, pp. 25–37, Mar. 2005.
- [2] A. Hampapur, L. Brown, J. Connell, A. Ekin, N. Haas, M. Lu, H. Merkl, S. Pankanti, A. Senior, C.-F. Shu, and Y.-L. Tian, “Smart video surveillance: exploring the concept of multiscale spatiotemporal tracking,” *IEEE Signal Process. Mag.*, vol. 22, no. 2, pp. 38–51, Mar. 2005.
- [3] M. Bramberger, A. Doblander, A. Meier, B. Rinner, and A. Schwabach, “Distributed embedded smart cameras for surveillance applications,” *IEEE Computer*, vol. 39, no. 2, pp. 68–75, Feb. 2006.
- [4] J. Konrad, “Videopsy: Dissecting visual data in space-time,” *IEEE Comm. Mag.*, vol. 45, no. 1, pp. 34–42, Jan. 2007.
- [5] A. Elgammal, D. Harwood, and L. Davis, “Non-parametric model for background subtraction,” in *Proc. European Conf. Computer Vision*, 2000.



(a) Original frames



(b) Motion detection results for spatially- and temporally-constant translation camera motion model



(c) Motion detection results for spatially-affine, time-varying camera motion model

Figure 3. Motion detection results for natural sequences *Foreman* and *Stefan* (top row) with significant camera pan using approach modeling camera motion as spatially- and temporally-constant translation (middle row) and spatially-affine, time-varying transformation (bottom row).

- [6] C. Stauffer and E. Grimson, "Learning patterns of activity using real-time tracking," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 22, no. 8, pp. 747–757, 2000.
- [7] Y. Sheikh and M. Shah, "Bayesian modeling of dynamic scenes for object detection," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 27, no. 11, pp. 1778–1792, 2005.
- [8] N. Paragios and R. Deriche, "Geodesic active contours and level sets for the detection and tracking of moving objects," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 22, no. 3, pp. 266–280, Mar. 2000.
- [9] S. Jehan-Besson, M. Barlaud, and G. Aubert, "DREAM²S: Deformable regions driven by an Eulerian accurate minimization method for image and video segmentation," *Intern. J. Comput. Vis.*, vol. 53, no. 1, pp. 45–70, 2003.
- [10] M. Ristivojević and J. Konrad, "Space-time image sequence analysis: object tunnels and occlusion volumes," *IEEE Trans. Image Process.*, vol. 15, no. 2, pp. 364–376, Feb. 2006.
- [11] T. Aach and A. Kaup, "Bayesian algorithms for adaptive change detection in image sequences using Markov random fields," *Signal Process., Image Commun.*, vol. 7, pp. 147–160, 1995.
- [12] F. Luthon, A. Caplier, and M. Liévin, "Spatiotemporal MRF approach with application to motion detection and lip segmentation in video sequences," *Signal Process.*, vol. 76, pp. 61–80, 1999.
- [13] V. Mezaris, I. Kompatsiaris, and M. Strintzis, "Video object segmentation using Bayes-based temporal tracking and trajectory-based region merging," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 14, no. 6, pp. 782–795, Jun. 2004.
- [14] F.-M. Porikli and Y. Wang, "An unsupervised multi-resolution object extraction algorithm using video-cube," in *Proc. IEEE Int. Conf. Image Processing*, 2001, pp. 359–362.
- [15] Y. Wang, J. Ostermann, and Y.-Q. Zhang, *Video Processing and Communications*. Prentice Hall, 2002.
- [16] R. Feghali and A. Mitiche, "Spatiotemporal motion boundary detection and motion boundary velocity estimation for tracking moving objects with a moving camera: A levels sets PDEs approach with concurrent camera motion compensation," *IEEE Trans. Image Process.*, vol. 13, no. 11, pp. 1473–1490, Nov. 2004.
- [17] C. Stiller, "Object-based estimation of dense motion fields,"

- IEEE Trans. Image Process.*, vol. 6, no. 2, pp. 234–250, Feb. 1997.
- [18] E. Mémin and P. Pérez, “Dense estimation and object-based segmentation of the optical flow with robust techniques,” *IEEE Trans. Image Process.*, vol. 7, no. 5, pp. 703–719, May 1998.
- [19] A.-R. Mansouri, A. Mitiche, and J. Konrad, “Selective image diffusion: application to disparity estimation,” in *Proc. IEEE Int. Conf. Image Processing*, vol. 3, Oct. 1998, pp. 284–288.
- [20] R. Feghali, A. Mitiche, and A.-R. Mansouri, “Tracking as motion boundary detection in spatio-temporal space,” in *Int. Conf. Imaging Science, Systems, and Technology*, Jun. 2001, pp. 600–604.
- [21] S. Jehan-Besson, M. Barlaud, and G. Aubert, “Detection and tracking of moving objects using a new level set based method,” in *Proc. Int. Conf. Patt. Recog.*, Sep. 2000, pp. 1112–1117.
- [22] B. Parker and J. Magarey, “Three-dimensional video segmentation using a variational method,” in *Proc. IEEE Int. Conf. Image Processing*, 2001, pp. 765–768.
- [23] J. Sethian, *Level Set Methods*. Cambridge University Press, 1996.
- [24] Y. Shi and W. Karl, “A fast level set method without solving PDEs,” in *Proc. IEEE Int. Conf. Acoustics Speech Signal Processing*, Mar. 2005, pp. 97–100.
- [25] —, “A real-time algorithm for the approximation of level-set-based curve evolution,” *IEEE Trans. Image Process.*, vol. 17, no. 5, pp. 645–656, May 2008.
- [26] S. Geman and D. McClure, “Statistical methods for tomographic image reconstruction,” in *Proc. 46th Session of the Int. Statistical Institute*, 1987, vol. 52.
- [27] M. Black, “Robust incremental optical flow,” Ph.D. dissertation, Yale University, Department of Computer Science, Sep. 1992.