Object Tracking by Segmentation Using Incremental Import Vector Machines

Ribana Roscher, Jan Siegemund, Falko Schindler, Wolfgang Förstner

TR-IGG-P-2012-01

13.04.2011

Technical Report Nr. 1, 2012

Department of Photogrammetry
Institute of Geodesy and Geoinformation
University of Bonn

Available at
http://www.ipb.uni-bonn.de/papers/

Object Tracking by Segmentation Using Incremental Import Vector Machines

Ribana Roscher, Jan Siegemund, Falko Schindler, Wolfgang Förstner

Zusammenfassung

We propose a framework for object tracking in image sequences, following the concept of tracking-by-segmentation. The separation of object and background is achieved by a consecutive semantic superpixel segmentation of the images, yielding tight object boundaries. I.e., in the first image a model of the object's characteristics is learned from an initial, incomplete annotation. This model is used to classify the superpixels of subsequent images to object and background employing graph-cut. We assume the object boundaries to be tight-fitting and the object motion within the image to be affine. To adapt the model to radiometric and geometric changes we utilize an incremental learner in a co-training scheme. We evaluate our tracking framework qualitatively and quantitatively on several image sequences.

1 Introduction

We propose a tracking approach based on a *tracking-by-segmentation* scheme.

In contrast to tracking-by-detection (e.g., [20], [5], [21], [11], [2]), where often only a bounding box or an ellipse is obtained, tracking-by-segmentation enables a tight object boundary. Recent work using the latter concept shows promising results, e.g., in fields like action recognition [25] and car tracking [6].

Several methods have been proposed treating tracking as binary classification of object and background. E.g., the authors of [26] and [21] use a discriminative model for segmentation, being a powerful model for dealing with background-clutter. Ren and Malik [18] use a conditional random field (CRF, [15]) combining an adaptively learned appearance model with the prior knowledge of a spatial model. Unger et al. [22] interpret tracking as segmentation in a spatial-temporal volume with 2D frames and the temporal domain as third dimension. Both off-line and incremental methods (e.g. [11], [26], [21]) have been used for learning the models for classification.

Other segmentation techniques represent the object as collection of features or its contour. E.g., the shape-based approach [10] uses MSER features [17], while the contour-based approach [13] uses edge features to segment the image. The support vector tracker [4] uses support vector machines [23] with edge-features to train a model and to integrate it into an optical-flow based tracker.

Another way to restrict the number of expensive computations within the tracking is to pre-segment the image into superpixels, representing small homogeneous image regions (e.g. [18], [24]).

Our contribution is to propose an object tracking framework providing a semantic superpixel segmentation of image sequences into object and background. For *efficient* computation we use SLIC superpixels [1] to narrow down the segmentation task to classify regions rather than single pixels. We consider *spatial as well as temporal relations* between superpixels, employing CRF and superpixel motion estimation. We use a *discriminative model* for tracking, since it can deal well with background clutter and needs no complex description of object and background. To be *adaptive* to radiometric and geometric changes in the images we use a learning method called incremental import vector machines (I^{2}VM, [3]), being an incremental formulation the import vector machines (IVM, [27]). To be *robust* and to ensure a reliable acquisition of new training samples we incorporate a large variety of features in a co-training scheme [7], i.e. appearance, motion and depth obtained from stereo images. We evaluate the tracking framework qualitatively and quantitatively on several image sequences.

The paper is organized as follows. In Section 2 we describe the incremental IVM and the co-training scheme. In Section 3 we propose our tracking framework. During our experiments in Section 4 we evaluate the tracking framework qualitatively and quantitatively. We summarize and conclude in Section 5.

2 Background

In this section we briefly introduce the off-line IVM and the I^{2}VM, which we use in our tracking framework. We also consider the co-training scheme, enabling a reliable acquisition of new labeled training samples to update the I^{2}VM model.

2.1 Import vector machines

Off-line model. Zhu and Hastie [27] proposed a probabilistic kernel-based discriminative algorithm for classification, the so-called Import Vector Machines. Following the idea of the SVM, they only choose a subset out of the training set, the import vectors, whose parameters defining the decision boundary are non-zero. The IVM are a realization of a sparse kernel logistic regression. The model parameters are optimized in a greedy procedure with simultaneous import vector selection. The off-line algorithm requires all training samples in advance to train the model and has to be re-trained if new training samples become available.

Incremental model. If data samples become available sequentially, e.g., as in tracking scenarios, it is reasonable and more efficient to update the IVM incrementally, rather than recomputing from scratch.

To adapt to radiometric and geometric changes, there are four steps for updating the model parameters: First we add new training vectors. In the second step we add import vectors that represent the current appearance of object and background. We remove import vectors not representing object and background anymore. Finally we remove training vectors to prevent the system from growing continuously. We refer to [3] for further details.

2.2 Co-training

To update the model obtained from the incremental learner, we need new labeled training samples. Starting from a few labeled samples we use a co-training scheme to label unlabeled data in each new frame. We independently train classification models $\mathcal{M}$ with different features f and combine the posterior probabilities obtained from each feature P_f directly with $P_{\mathcal{M}} = \frac{1}{Z}\prod_f P_f$ and Z being the normalization function. We choose the most confident predictions as new labeled training samples. If one feature is not sufficient to accurately predict the image segmentation, e.g., if the feature is not discriminating, other features can still provide a reliable prediction. The co-training technique works best if the features are complementary, e.g., like motion, geometry and appearance.

3 Our proposed tracking framework

In this section we explain our proposed tracking framework in detail. We also consider the extraction and tracking of superpixels and the formulation of the CRF model. In Section 3.1 we give an overview of the tracking framework. In the subsequent sections the single steps are referred to in more detail.

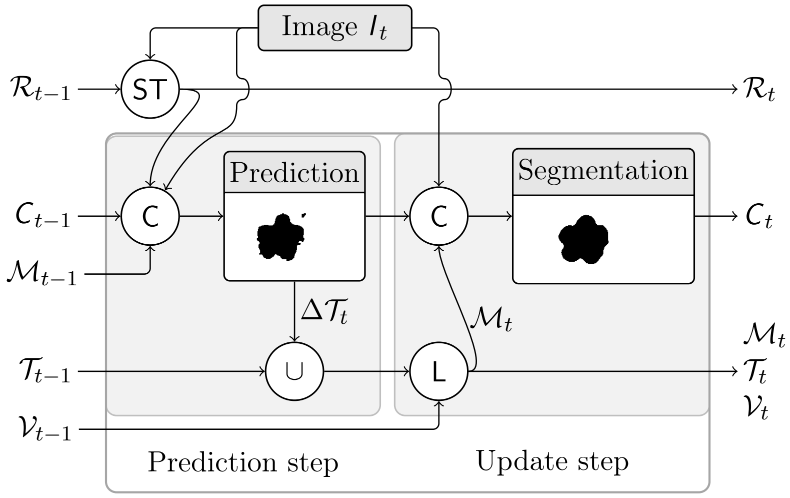

Abbildung 1: Using tracked superpixel regions $\mathcal{R}_t$, the classification C_{t-1}, the posteriors P_{t-1} and models $\mathcal{M}_{t-1}$ obtained from the incremental learning procedure, we predict a segmentation C'_t. We update the set of labeled training samples $\mathcal{T}$ and learn models $\mathcal{M}_t$ to derive the final segmentation C_t with posteriors P_t.

3.1 Tracking framework

Our tracking framework is schemed in Figure 1. The extraction of superpixels $\mathcal{R}$ in the image I_t at a given time step t and the tracking of superpixels from $t-1$ to t is explained in Section 3.2 and denoted with ST. If a stereo image pair $\mathcal{I}_t$ is given, we refer to the left image as reference image. Furthermore we have given a set of models $\mathcal{M}_t$, which consists of all trained models, one for each feature f.

Our tracking framework is based on a prediction and an update step, both containing a classification C. The prediction step is used to acquire new labeled training data, enabling the learned models $\mathcal{M}_t$ to adapt to radiometric and geometric changes in the update step.

Initialization. We start from an initial, user-defined, incomplete segmentation and train the first models $\mathcal{M}_1$ with the off-line IVM. The output are posterior probabilities $P_{\mathcal{M}_1}$ of the features extracted in image I_1. The classification C yields a segmented image C_1. Only the most discriminating features are identified and considered for the tracking scheme.

Prediction step. In the prediction step we use priors P_B and P_S for the object's geometric properties, obtained from the segmented image C_{t-1} and the posteriors P_{t-1}, as well as the models $\mathcal{M}_{t-1}$ to predict a new segmented image C'_t. The derivation of these priors is explained in Section 3.4. From the predicted segmentation C'_t we sample the most confident predictions as new labeled data, i.e. superpixels, and use it for the update step.

Update step. In the update step we incrementally learn our new models $\mathcal{M}_t$ in a co-training scheme (Section 2.2), denoted with L. With the new models and the priors P_S and P_B we obtain posteriors P_t. The classification C yields a final segmentation C_t.

3.2 Superpixel computation

The usage of superpixels has become popular in several applications with the need of feature extraction. They catch redundancy in the image and reduce the complexity to train and to test classifiers.

Superpixel extraction. We follow the idea of [1] to generate superpixels that are compact, have a regular shape, but are also homogeneous in their spectral features. Pixels $i = 1 \ldots I$ are clustered to superpixels $k \in \mathcal{R} = \{1 \ldots K\}$ based on their spectral appearance $\boldsymbol{f}_i$ (Lab-color vector) and position in the image $\boldsymbol{p}_i$.

The algorithm is initialized by sampling K cluster centers from the image at a regular grid interval of g pixels. Each pixel in the image is then assigned to one cluster center $\boldsymbol{c}_k$ using the K-means-algorithm, where the distance of pixels and cluster centers is computed by the similarity measure $d_{ki} = ||\boldsymbol{c}_k - \boldsymbol{c}_i||_2$ with $\boldsymbol{c}_{(\cdot)} = [\boldsymbol{f}_{(\cdot)}, \frac{w_\text{S}}{g}\boldsymbol{p}_{(\cdot)}]^\mathsf{T}$ [1]. The parameter w_S weights the influence of spatial proximity. The higher w_S is, the more compact are the superpixels. The grid interval g adapts this weight to the image resolution.

Superpixel tracking. In order to keep the affiliation of image structures to superpixels over time, we extend this approach to a superpixel tracking scheme similar to the one presented in [16].

The basic idea is to predict the position of the superpixel centers $\boldsymbol{p}_{k,t-1}$ in the subsequent frame t, using the optical flow information of its assigned pixels. The predicted centers $\widetilde{\boldsymbol{p}}_{k,t}$ are then used as initial cluster centers for the superpixel segmentation of frame t. For dense optical flow computation, we employ the approach of [9].

In contrast to [16], where the displacement of the center positions $\boldsymbol{p}_{k,t-1}$ and $\boldsymbol{p}_{k,t}$ is computed from the weighted average of the flow vectors of all pixels assigned to k, we predict the new cluster centers $\widetilde{\boldsymbol{c}}_{k,t}$ using a Kalman filter. We assume a constant affine motion of the image region covered by one superpixel as well as invariance in its spectral appearance.

Further, we introduce a birth and death process to handle occlusions and recently discovered regions: Superpixels exceeding a maximum area are split into two successors, while those falling below a minimum area are merged to one of their neighbors. This birth and death process is illustrated in Figure 2.

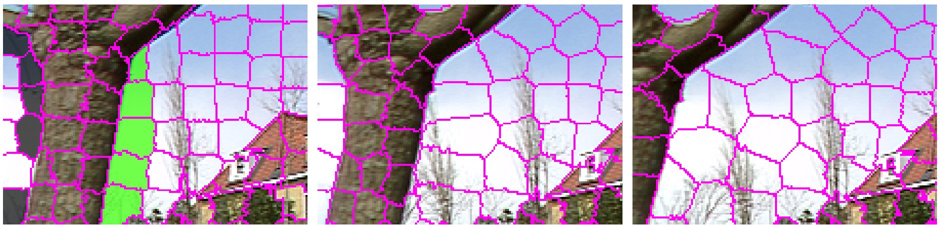

Abbildung 2: Example sequence from the Flower garden data set demonstrating the birth and death process for superpixel tracking. While the tree is moving to the left, some background regions are occluded and others are rediscovered. The green highlighted regions in the left image are examples for superpixels to be split and the dark highlighted ones are merged to their neighbors.

3.3 Features for object representation

For each superpixel segment in $\mathcal{R}_t$ we extract a set of features $\mathcal{X}$ from (1) radiometric appearance: mean and standard deviation of RGB/HSV/Lab color, (2) motion: mean optical flow, and (3) geometry: mean position and mean disparity (if stereo images are available).

To obtain a disparity measurement for nearly every pixel within the image, we employ a dense stereo approach [12]. Pixels with no disparity information, e.g., caused by stereo shadow, are flagged as invalid and are not considered for computing the mean disparity feature.

For optical flow the motion information is extracted using [9], yielding a displacement vector for every pixel.

3.4 Classification

To incorporate prior knowledge about the spatial and temporal relations between tracked superpixels we model our task as a CRF, as shown in Figure 3. We prefer short object boundaries and temporal consistency of the superpixel labels.

Model. Our CRF model is defined as

$$\begin{aligned} E_t(\boldsymbol{y}_t) = &- \sum_{k \in \mathcal{R}_t} \log \left(P_t \left(y_{k,t} | \mathcal{X}_{k,t} \right) \right) + w_{\text{temp}} \sum_{k \in \mathcal{R}_t} \Psi \left(y_{k,t}, \widehat{y}_{k,t-1}, \boldsymbol{c}_{k,t}, \widetilde{\boldsymbol{c}}_{k,t} \right) \\ &+ w_{\text{dis}} \cdot w_{\text{spatial}} \sum_{(k,k') \in \mathcal{N}_t} \Phi \left(y_{k,t}, y_{k',t}, \mathcal{X}_{k,t}, \mathcal{X}_{k',t} \right). \end{aligned} \tag{1}$$

The labels of the superpixels are given by y_k. The first, unary term is defined as the negative logarithm of the posteriors P_t obtained in the co-training

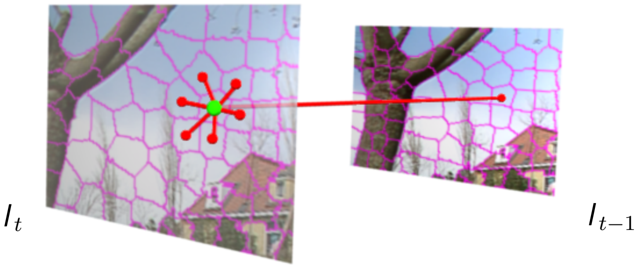

Abbildung 3: Spatial and temporal relations between the superpixels of two frames $t-1$ and t. Each superpixel is connected to its neighbors within the image and to its neighbor from the previous frame, assigned via superpixel tracking.

scheme. The second, unary term given by the function Ψ is a temporal consistency term. Since we assume the final labeling of the superpixels to be smooth within the image, we introduce this prior knowledge by means of a data-depended Potts model in the third, binary term. The set of spatial neighbors is denoted by $\mathcal{N}_t$.

We use graph-cut[1] [8] to solve for the best labeling $\widehat{\boldsymbol{y}}_t = \operatorname{argmin}_{\boldsymbol{y}_t} E_t(\boldsymbol{y}_t)$. To obtain posteriors P_t, we use local marginalization.

The weighting parameters w_{temp} and w_{spatial} are set empirically via cross-validation. We also introduce a weight w_{dis} to ensure a reliable weighting of the unary, co-training based term and the binary term. If the features are only little discriminative, the spatial term gets a low weight preventing the binary term from getting too dominant. The weighting parameter w_{dis} is computed in each time step: $w_{\text{dis}} = 1 - 4(\max P_t)(1 - \max P_t) \in \,]0, 1[$.

Unary, co-training-based terms. To obtain a reliable classification and sampling of new data we introduce two priors: P_B and P_S.

Depending on the object's position in the previous frame, we define a bounding box around the slightly dilated object. I.e., the bounding box prior P_B is expressed as a hard assignment of probability 0 to locations far away from the object's previous position.

The prior P_S arises from the optical flow: Given the superpixel correspondences from image I_{t-1} to I_t, we obtain the P_S by transferring the posteriors P_{t-1} from the previous frame to the current time step t. The final posterior probability is given by

$$P_t = \frac{1}{Z} P_{\mathcal{M}} \cdot (P_S)^w \cdot P_B. \tag{2}$$

The weight w is chosen depending on the reliability of the flow P_S. When the object got lost during the sequence, w is set to 0.

[1] http://vision.csd.uwo.ca/code/

Unary temporal terms. The unary temporal term is modeled as the similarity of the superpixel's current feature vector $\boldsymbol{c}_{k,t}$ and its predicted feature vector $\widetilde{\boldsymbol{c}}_{k,t}$ assigned via superpixel tracking. If the distance is small, it is likely that both superpixels belong to the same class, and vice versa. The similarity is defined as the cosine of the angle between the feature vectors $\psi_{k,t} = \cos\angle(\boldsymbol{c}_{k,t}, \widetilde{\boldsymbol{c}}_{k,t})$. The unary temporal term is

$$\Psi\left(y_{k,t}, \widehat{y}_{k,t-1}, \boldsymbol{c}_{k,t}, \widetilde{\boldsymbol{c}}_{k,t}\right) = \begin{cases} 1 - |\psi_{k,t}| & , y_{k,t} = \widehat{y}_{k,t-1} \\ |\psi_{k,t}| & , y_{k,t} \neq \widehat{y}_{k,t-1} \end{cases} \tag{3}$$

Binary spatial terms. The binary term is modeled as the normalized length of the border $l_{kk'}$, only considered if two superpixels k and k' get different labels: $\phi_{kk'} = l_{kk'}/\hat{l}_t$. The normalization factor $\hat{l}_t$ is estimated in each frame t as mode of the beta-distribution of all border lengths $l_{kk'}$.

4 Experiments

Data sets. First we perform a qualitative analysis with the data sets Galloping horse[2], Flower (stereo), Flower garden[3] and Scrat[4] (first four rows in Figure 5). They are challenging due to moving objects, changing illumination, varying object appearance, size and shape, changing or moving background, and/or occlusions. We also perform a quantitative evaluation on the data sets Lemming, Box, Board and Liquor with given ground truth rectangles[5]. We compare our approach to five tracking methods ([20], [5], [2], [19], [14]).

Results. The objects are segmented correctly, even when radiometric or geometric characteristics change or the object disappears. The Flower data set shows similar results when processed with appearance in combination with either motion or disparity. For Galloping horse and Flower garden a high spatial weight shows best results. For the fast moving object in the Scrat data set the best result was achieved with $P_S = 0$ and small spatial and temporal weights.

Figure 4 demonstrates the influence of the spatial and temporal terms. Omitting the spatial term provides worse results, since some superpixels close to the boundary with similar features are false classified. When omitting the

[2] http://www.cs.toronto.edu/~babalex/SpatiotemporalClosure/supplementary_material.html

[3] http://persci.mit.edu/demos/jwang/garden-layer/orig-seq.html

[4] http://www.iceagemovie.com/

[5] http://gpu4vision.icg.tugraz.at/index.php?content=subsites/prost/prost.php

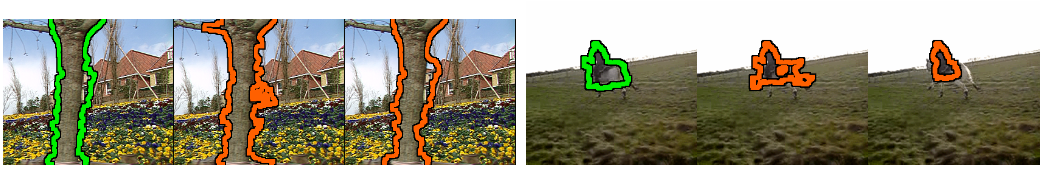

Abbildung 4: Example sequence from the Flower garden and Galloping horse data sets demonstrating the influence of the spatial and the temporal term. The left image of each data set is the result incorporating both terms, the middle image without spatial term and the right image without temporal term.

temporal term parts of the object are lost and the segmentation is less smooth over time.

Algorithm	Overall	Board		Box		Lemming		Liquor	
		pascal	dist	pascal	dist	pascal	dist	pascal	dist
PROST [20]	80.4	75.0	39.0	90.6	13.0	70.5	25.1	85.4	21.5
MILTrack [5]	49.2	67.9	51.2	24.5	104.6	83.6	14.9	20.6	165.1
FragTrack [2]	66.0	67.9	90.1	61.4	57.4	54.9	82.8	79.9	30.7
ORF [19]	27.3	10.0	54.5	28.3	145.4	17.2	166.3	53.6	67.3
GRAD [14]	88.9	94.3	14.7	91.8	13.2	78.0	28.4	91.4	11.9
I^{2}VM	55.0	89.8	17.7	25.4	143.3	72.7	16.1	31.9	53.2
I^{2}VM (fixed)	71.9	95.1	15.8	35.7	143.2	86.1	16.8	68.7	53.9

Tabelle 1: Pascal distance and distance score in the Lemming, Box, Board and Liquor sequences in comparison to five tracking methods. Distance score is the average euclidean distance between the centers of the tracking rectangle and the ground truth rectangle. Our tracking rectangle is the bounding box of our segmentation. In the last row we evaluate a fixed-size bounding box centered on the centroid of our segmentation. The pascal distance is the percentage of frames, where the overlapping area of the tracking rectangle and the ground truth rectangle exceeds 50 %.

Table 1 shows the results of our proposed tracking approach with I^{2}VM in comparison to five other methods that use tracking-by-detection with a fixed-size bounding box.

We achieve a high accuracy in the Lemming and Board sequences dealing with various appearance variations, occlusions and different illumination.

In the Box sequence the object is lost after one third of the sequence, because the background has similar appearance and the flow used as features is not discriminating enough if the object is slow-moving. In almost the same manner, in some frames of the Liquor sequence the bottle could segmented only partly. Nevertheless our algorithm tracks the correct bottle most of the time.

Discussion. Our experiments show that our object tracking framework can handle occlusion and changes in appearance and geometry. Furthermore, it provides a detailed segmentation and not only a bounding box, making the algorithm versatile, e.g., for action recognition.

The computational bottlenecks of our tracking framework are the incremental learning and the superpixel tracking. Both of them can be parallelized. The current Matlab/C++ implementation takes about 1 second on an Intel(R) Dual Core with 3.0 GHz for the incremental learning. The extraction and tracking of 1000 superpixels takes about 1 second.

5 Conclusion

We proposed a tracking framework, which is based on solving the tracking task as semantic superpixel segmentation yielding a tight object boundary. We showed in our experiments that our approach can handle occlusion, moving backgrounds, changes in shape, size and appearance of the object. The incorporation of spatial and temporal relations between the superpixels could improve the segmentation results. A quite promising extension would be to track the superpixels of the object as ensemble and to update object and background coherently. Furthermore it is to be investigated how the classification results can assist superpixel extraction.

Literatur

[1] R. Achanta, A. Shaji, K. Smith, A. Lucchi, P. Fua, and S. Süsstrunk. SLIC Superpixels. Technical report, EPFL, 2010.

[2] A. Adam, E. Rivlin, and I. Shimshoni. Robust Fragments-based Tracking Using the Integral Histogram. In *CVPR*, pages 798–805, 2006.

[3] Authors. Incremental Import Vectors Machines for Classification and Object Tracking. In *ICCV*, 2011. ICCV 2011 Submission ID 248, Supplied as additional material incrementalIVM.pdf.

[4] Shai Avidan. Support Vector Tracking. *PAMI*, 26:1064–72, 2004.

[5] B. Babenko, M.H. Yang, and S. Belongie. Visual tracking with online multiple instance learning. In *CVPR*, pages 983–990, 2009.

[6] A. Barth, J. Siegemund, A. Meißner, U. Franke, and W. Förstner. Probabilistic Multi-Class Scene Flow Segmentation for Traffic Scenes. In *DAGM*, 2010.

[7] A. Blum and T. Mitchell. Combining Labeled and Unlabeled Data with Co-training. In *COLT*, pages 92–100, 1998.

[8] Y. Boykov, O. Veksler, and R. Zabih. Fast Approximate Energy Minimization via Graph Cuts. *PAMI*, 23:2001, 2001.

[9] A. Chambolle and T. Pock. A First-order Primal-dual Algorithm for Convex Problems with Applications to Imaging. *preprint.*

[10] M. Donoser and H. Bischof. Efficient Maximally Stable Extremal Region (MSER) Tracking. In *CVPR*, pages 553–560, 2006.

[11] H. Grabner and H. Bischof. On-line Boosting and Vision. In *CVPR*, pages 260–267, 2006.

[12] H. Hirschmüller. Accurate and Efficient Stereo Processing by Semi-global Matching and Mutual Information. In *CVPR*, pages 807–814, 2005.

[13] M. Isard and A. Blake. Contour Tracking by Stochastic Propagation of Conditional Density. In *ECCV*, pages 343–356, 1996.

[14] A. Klein and A. B. Cremers. Boosting Scalable Gradient Features for Adaptive Real-Time Tracking. In *ICRA*, 2011.

[15] J. Lafferty, A. McCallum, and F. Pereira. Conditional Random Fields: Probabilistic Models for Segmenting and Labeling Sequence Data. In *Machine Learning*, pages 282–289, 2001.

[16] A. Levinshtein, C. Sminchisescu, and S. Dickinson. Spatiotemporal Closure. In *ACCV*, 2010.

[17] J. Matas, O. Chum, M. Urban, and T. Pajdla. Robust Wide-Baseline Stereo from Maximally Stable Extremal Regions. *Image and Vision Computing*, 22(10):761–767, 2004.

[18] X. Ren and J. Malik. Tracking as Repeated Figure/Ground Segmentation. In *CVPR*, pages 1–8, 2007.

[19] A. Saffari, C. Leistner, J. Santner, M. Godec, and H. Bischof. On-line Random Forests. In *ICCV Workshops*, pages 1393–1400, 2009.

[20] J. Santner, C. Leistner, A. Saffari, T. Pock, and H. Bischof. PROST: Parallel Robust Online Simple Tracking. In *CVPR*, pages 723–730, 2010.

[21] F. Tang, S. Brennan, Q. Zhao, and H. Tao. Co-Tracking Using Semi-Supervised Support Vector Machines. In *ICCV*, pages 1–8, 2007.

[22] M. Unger, T. Mauthner, T. Pock, and H. Bischof. Tracking as Segmentation of Spatial-Temporal Volumes by Anisotropic Weighted TV. In *EMMCVPR*, pages 193–206, 2009.

[23] V.N. Vapnik. *The Nature of Statistical Learning Theory*. Springer Verlag, 2000.

[24] A. Vazquez-Reina, S. Avidan, H. Pfister, and E. Miller. Multiple Hypothesis Video Segmentation from Superpixel Flows. In *ECCV*, pages 268–281, 2010.

[25] D. Weinland, E. Boyer, and R. Ronfard. Action Recognition from Arbitrary Views using 3D Exemplars. In *ICCV*, pages 1–7, 2007.

[26] Q. Yu, T. Dinh, and G. Medioni. Online Tracking and Reacquisition using Co-trained Generative and Discriminative trackers. In *ECCV*, pages 678–691, 2008.

[27] J. Zhu and T. Hastie. Kernel Logistic Regression and the Import Vector Machine. *Computational and Graphical Statistics*, 14:185–205, 2005.

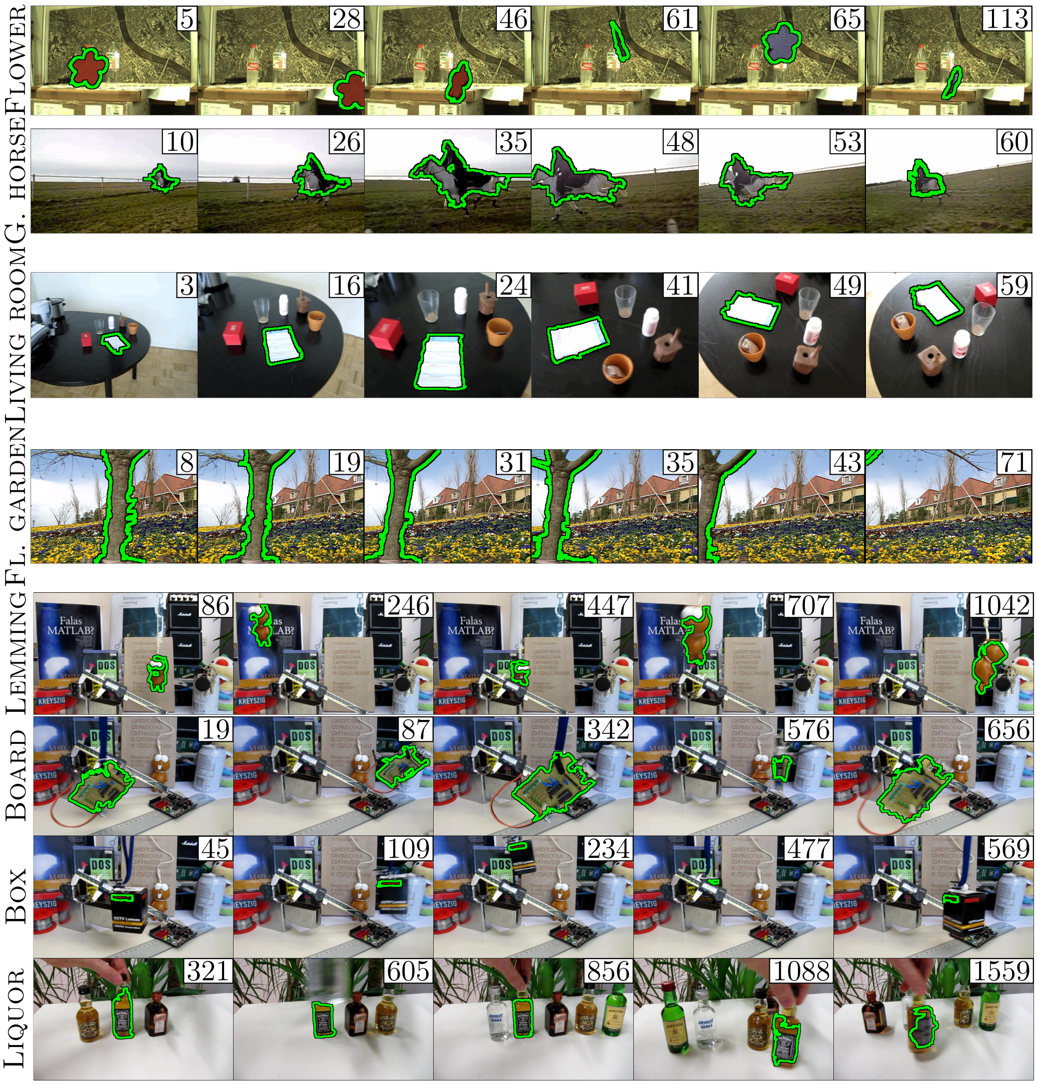

Abbildung 5: Representative frames (numbers t in the top right corners) of the tracking results for the evaluated sequences.