Markovian Tracking-by-Detection from a Single, Uncalibrated Camera

Michael D. Breitenstein[1] Fabian Reichlin[1] Bastian Leibe[1,2] Esther Koller-Meier[1] Luc Van Gool[1,3]
[1]ETH Zurich [2]RWTH Aachen [3]KU Leuven

Abstract

We present an algorithm for multi-person tracking-by-detection in a particle filtering framework. To address the unreliability of current state-of-the-art object detectors, our algorithm tightly couples object detection, classification, and tracking components. Instead of relying only on the final, sparse output from a detector, we additionally employ its continuous intermediate output to impart our approach with more flexibility to handle difficult situations. The resulting algorithm robustly tracks a variable number of dynamically moving persons in complex scenes with occlusions. The approach does not rely on background modeling and is based only on 2D information from a single camera, not requiring any camera or ground plane calibration. We evaluate the algorithm on the PETS'09 tracking dataset and discuss the importance of the different algorithm components to robustly handle difficult situations.

1. Introduction

Multi-people tracking plays an important role in various computer vision applications, such as surveillance, sports video analysis, traffic control, and robot navigation. A tracking algorithm aims to continuously estimate the positions of a variable number of targets in a scene over time. The resulting trajectories can then provide the basis for scene understanding, for example to automatically recognize and interpret the behavior of agents.

Typically, two major components can be distinguished in tracking algorithms: A *bottom-up* process deals with target representation and localization, trying to cope with changes in the appearance of the tracked targets, while a *top-down* process performs data association and filtering to deal with object dynamics. Correspondingly, our approach is based on combining a state-of-the-art pedestrian detector (bottom-up) with particle filtering (top-down). To complement the generic object category knowledge from the detector, our algorithm trains instance-specific classifiers online, which are used to evaluate each detection-target pair for data association in each frame. Our method is based only on 2D information from one single camera and does not require any camera or ground plane calibration.

In contrast to background modeling-based trackers, tracking-by-detection methods are generally robust to moving cameras and have therefore become increasingly popular [2, 9, 14, 15, 22]. Such approaches involve the continuous application of a detection algorithm in individual frames and the association of detections across frames. The main challenge is to deal with the unreliability of the sparse detector output, because object detectors typically deliver only a discrete set of responses and usually yield false positives and miss detections.

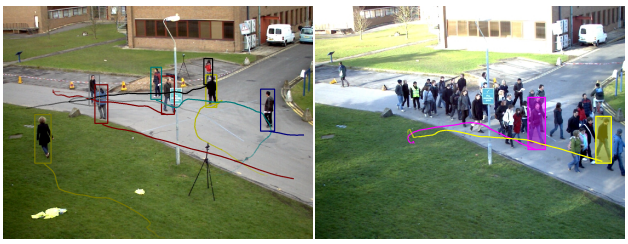


Figure 1: Tracking results for the PETS'09 datasets.

While many recent tracking-by-detection approaches address the resulting data association problem by optimizing detection assignments over a larger temporal window in an offline step [1, 3, 10, 14, 16], particle filters offer a framework for representing the tracking uncertainty in a Markovian manner. Such an approach is more suitable for time-critical, online applications, because only information from past frames is used.

Previous methods for multi-object tracking-by-detection rely only on the final, sparse output from an object detector [5, 8, 15, 22]. In contrast, our approach integrates the detector into the tracking process by monitoring its continuous detector confidence, which we use as a graded observation model. This idea follows the intuition that by avoiding hard detection decisions, we can impart our tracking approach with more flexibility to handle difficult situations. Furthermore, our method evaluates detection-target pairs for data association not only based on the spatial distance like previous methods [5, 15], but based on the output of classifiers that are trained online (similar to Song *et al.* [19], which is however based on background modeling).

In this paper, we demonstrate the potential of our Markovian tracking-by-detection algorithm, which is based only on 2D information from a single, uncalibrated camera, by

evaluating it on the PETS'09 tracking datasets (S2, view 001, see Fig. 1). A detailed discussion illustrates the influence and limitations of each algorithm component.

2. Related Work

In recent years, a vast amount of work on multi-object tracking has been presented. Generally, the methods can be divided either in *global* approaches that aim to construct trajectories offline after *all* observations are received, or in *local, Markovian* approaches that estimate the position of a target by considering only information from the past (*e.g.*, from the *last* time step).

Tracking algorithms based on background modeling (*e.g.*, [19, 20]) are usually capable of online processing. By combining foreground information from multiple views, they can be made more robust to occlusions [3, 12]. However, the cameras need to be static and the methods rely on ground plane or camera calibration. In contrast, our approach is based only on 2D information from one single, potentially moving, uncalibrated camera.

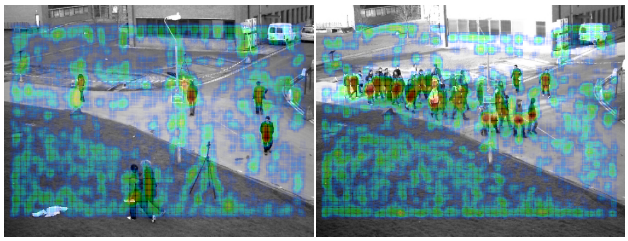
The state space uncertainty of a target can be represented using particle filters [11]. Later extensions include a representation of the joint state space for multiple targets [21] and the combination with an object detector for *tracking-by-detection* [8, 15]. As runtime directly scales with the number of particle evaluations, those approaches face a dilemma when additional targets appear. They can either spend an exponentially growing number of particles on representing the joint state space sufficiently well, or they can guarantee a constant runtime by keeping the number of particles fixed, at the price of lowering approximation accuracy. This can be solved by using independent particle sets for each target [5], at the cost of potential problems with occlusions.

Using independent trackers requires solving a data association problem to assign measurements to multiple targets. Classical approaches for this include Joint Probabilistic Data Association Filters (JPDAF) [7, 18] and Multi-Hypothesis Tracking (MHT) [17]. While some tracking-by-detection algorithms implement a first-order Markovian approach to perform data association based only on previous observations from the last time step [5, 8, 15, 22], several recent methods address the problem globally by optimizing detection assignments over a larger temporal window [1, 10, 14, 16].

3. Tracking using the Detector Output

Our approach is based on using the output of an object detector for the observation model of a particle filter. A general problem with this is the reliability of the resulting detections; *i.e.*, not all persons are detected in each frame (*missing detections*) and some detections are not caused by a person (*false positive detections*). This can be seen

(a) Final HOG Detections.



(b) Intermediate Output: Continuous Detector Confidence Density.

Figure 2: *(a)* The final output of an object detector (HOG [6]) contains false positives and missing detections. *(b)* The continuous detector confidence density (shown as heat map) often contains useful information at the location of missing detections, which we use for tracking.

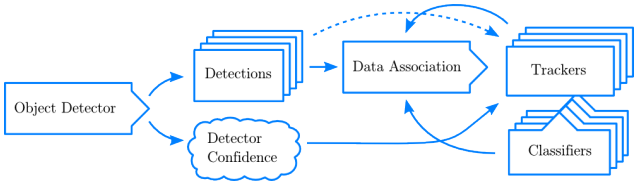


Figure 3: Our algorithm achieves robustness through a careful interplay between object detection, classification, and target tracking components (see text).

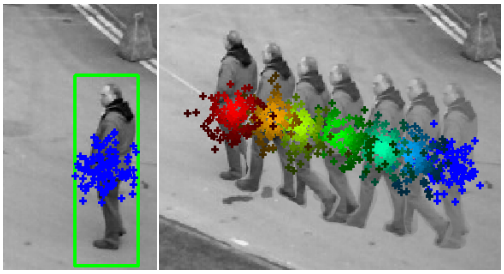
in Fig. 2(a), where incorrect detections are often found on background structures and around the road sign in the center of the image. Note also the different sizes of the hypothesized detections. Because no 3D or scene information (*e.g.*, ground plane) is available, the detector does not know where to expect objects of which size in the image.

Our algorithm addresses this uncertainty through a careful interplay between object detection, classification, and target tracking components. In the following, we shortly describe our approach (see Fig. 3 for an overview), referring to [4] for details.

Particle Filter. As a basic framework, we use a particle filter for each target, which estimates the target's time-evolving posterior distribution with a weighted set of particles. Each particle $\boldsymbol{x} = \{x, y, u, v\}$ encodes the 2D image position (x, y) and the velocity components (u, v).

We automatically initialize a particle filter for each tar-

Figure 4: The initialization and termination region of our algorithm for the PETS'09 S2.L1 dataset (view 001), shown in red.



(a) Initialization. (b) Propagation.

Figure 5: The initial particles (*a*) are drawn from a Normal distribution centered at the detection bounding box and are propagated (*b*) with a constant velocity motion model.

get in a certain image region (see Fig. 4). To be initialized, a target must be detected with similar position and size in two out of three subsequent frames and must neither be occluded by more than 20% nor associated with an already existing tracker. Similarly, a tracker only survives for $k = 5$ frames without associated detection in this region and is automatically terminated otherwise.

The particles are propagated with a constant-velocity motion model. Their weight is updated using the observation model described in the following (for details about particle filtering itself, we refer to the extensive literature).

Observation Model. The conditional likelihood $p(y_t|\boldsymbol{x}_t^{(i)})$ of a new observation of a particle i is estimated using three independent terms. In Sec. 4, we demonstrate the importance of each term for different situations. First, the *detection term* guides the particles of one tracker based on maximally one (carefully selected) associated detection per frame (see next paragraph). If a matching detection is found, this term robustly guides the particles. Otherwise, it will have no influence.

Although an object detector might not yield a (matching) final detection for every target, a tracking algorithm could still be guided using the intermediate output common to current state-of-the-art detectors. This can be seen in Fig. 2(b), where the intermediate output of the HOG detector is shown to have a high density on some persons, although no final detection is found. For sliding-window based detectors (*e.g.*, HOG [6]), such an intermediate output consists of the continuous confidence density, which is accumulated before applying non-maximum suppression. For feature-based detectors (*e.g.*, ISM [13]), it corresponds to the local voting space density. The *detector confidence density term* of the observation model estimates the detector confidence density at the particle position.

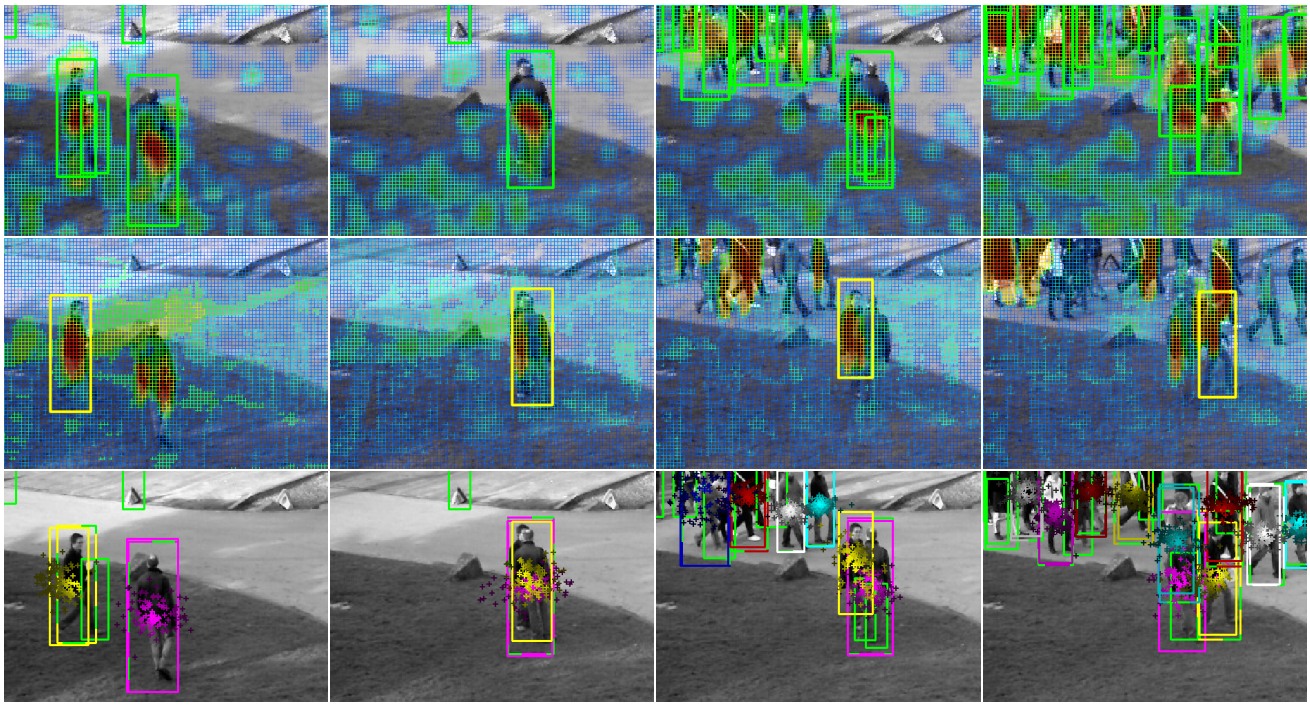
Finally, the particles are additionally weighted by evaluating the bounding boxes represented by each particle using target-specific classifiers. For this purpose, a classifier is trained online for each target (see next paragraph). In contrast to the other terms, this *classifier term* uses color information, complementing the other terms that are based on the detector output. This term adds additional robustness to our approach when targets are only partially visible, and it prevents the particle filter from drifting.

Data Association. One of the key factors of our algorithm is that a final detection is only used to guide a particle filter if it is very likely to belong to the respective tracking target. For this purpose, each detection-target pair is evaluated in each frame, and our algorithm assigns at most one detection to at most one tracker. A matching score matrix describes the match between each detection and target, consisting of the distance between the detection and each particle of a tracker and the score of a classifier trained online for each target (see below). Then, a greedy algorithm iteratively selects the pair with maximum score until no further valid pair is available. Finally, only the associated detections with a matching score above a threshold (set experimentally) are used, ensuring that a selected detection is actually a good match to a target.

The classifier consists of a boosted ensemble of weak learners (similar to [9]), containing color histograms (red-green-intensity, 3 bins per color channel) and local binary pattern features. Patches used as positive training examples are sampled from the bounding box of the associated detection. The negative training set is sampled from nearby targets, augmented by background patches. After each update step, we keep the 50 most discriminative weak learners.

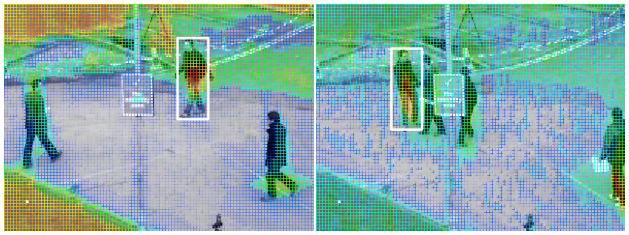
4. Detailed Implementation

Algorithm Parameters. The initial sample positions are drawn from a Normal distribution (with standard deviation $\sigma = 6 \cdot scale_{det}$ pixels), centered at the detection bounding box center (see Fig. 5). The initial target size corresponds to the size of the detection, where $scale_{det}$ is the factor of the size compared to the detector training size (48×96 pixels



(a) S2.L3, Frame 18 (b) S2.L3, Frame 105 (c) S2.L3, Frame 151 (d) S2.L3, Frame 175

Figure 7: Visualization of the different observation model terms. *Top:* Final HOG detections (green) and intermediate detector confidence density (heat map). *Middle:* Classifier output for the *yellow* tracker (heat map). *Bottom:* Particle output for all targets, together with mean modes and associated detections (dashed with the respective color). *Best viewed in color.*



(a) S2.L1, Frame 10 (b) S2.L1, Frame 39

Figure 6: Heat map visualization of the classifier response for the white target. The output becomes more discriminative over time because the classifier is continuously adapted (compare the confidence on the background in *a* and *b*).

total, of which the person itself takes up 24×72 pixels). The input images are resized from originally 768×576 pixels to 1280×960 pixels, such that the size of the persons better corresponds to the detector training size. The initial motion direction is set to be orthogonal to the closest image border (with magnitude $v = 24 \cdot scale_{det}$ pixels).

In each frame, the process noise for each state variable is independently drawn from zero-mean Normal distributions. The standard deviations for the position and velocity noise are set to $\sigma = 4 \cdot scale_{det}$ and $\sigma = 12 \cdot scale_{det}$ pixels, respectively (*i.e.*, about 10 and 30 pixels for a target with a height of 180 pixels ($scale_{det} = 2.5$)). The size of a tracker is set to the average of the last four associated detections. To determine the current position of a target (represented by the multi-modal particle distribution), the strongest mode is found using mean-shift.

Classifier Training. The classifiers trained online for one target result in a high score on the trained target itself and relatively low scores on the background and the other tracking targets. Because a classifier is only updated when a detection is assigned to the tracking target that furthermore must not be occluded by another target, the classifiers are becoming increasingly discriminative (see Fig. 6). Additionally, we use the bounding boxes corresponding to the mean modes of the nearest other trackers (maximally 6) and background information as negative training samples for the update. The classifiers help resolve ambiguities, especially in the case of inter-person occlusions (*e.g.*, see Figs. 6(b) and 7), and they complement the detector output to guide the particles.

Terms of the Observation Model. The balance between the different observation model terms is chosen such that the ratio is about $20 : 2 : 1$ between detection term,

detector confidence term and classifier term for a tracker with associated detection. During a typical tracking cycle, the contribution of each of the individual observation model terms to the total particle weight can however differ significantly. On average, a detection is selected and associated to a tracking target in about every other frame for S2.L1 and in every fourth frame for S2.L2 and S2.L3.

In Fig. 7, we illustrate how the different complementary observation model terms help to robustly handle a difficult situation where a target is occluded and the detections are unreliable. In Fig. 7(a), the yellow and magenta trackers have been initialized already and detections are associated in the current frame (bottom row; associated detections are dashed with the color of the respective tracker). Also, the detector confidence density is high on both targets (top row). The classifiers return high scores on the trained targets, but also moderate values on the background and the other targets (middle row; classifier output for the yellow tracker). The particles are weighted according to the observation model (bottom row; the weights are proportional to the color intensity). In the frame of Fig. 7(a), the detections are very good (*i.e.*, have a high confidence) and are used to primarily guide the particle filters.

In Fig. 7(b), the yellow tracker is almost fully occluded by the magenta tracker. Hence, our data association algorithm correctly does not associate a detection to the yellow tracker (Fig. 7(b), bottom row). The classifier output is low because only a part of the target the classifier is learned for is visible (middle row). However, because of the detector confidence density, the particles remain in about the same image region. In this situation, the particles are mainly guided by the detector confidence density.

Later, the target of the yellow tracker becomes partially visible again (see Fig. 7(c)). Still, the detector cannot accumulate enough evidence to detect the person (top row). However, since the classifier output is very high on the visible part of the target (middle row), the particles are concentrated correctly on the correct, partially occluded target.

When more people appear (see Fig. 7(d)), the classifier of the yellow tracker does not perfectly distinguish between all persons, because their appearance is similar (middle row). Therefore, a pure classifier-based tracking algorithm (*e.g.*, [2, 9]) would probably fail here, resulting in identity switches. However, the HOG object detector returns a good detection again that matches well with the learned appearance of the target of the yellow tracker (bottom row).

In contrast, the magenta tracker is guided by final detections through the frames of Fig. 7, but relies on the detector confidence density and the classifier terms in Fig. 7(d) (the classifier output for the magenta tracker is not shown).

5. Evaluation

We evaluate our algorithm for the tracking tasks S2.L1, S2.L2 and S2.L3, using the sequences from view 001 of the PETS'09 dataset.[1]

5.1. PETS Tracking Task S2.L1

All targets in the sequence S2.L1 are found and tracked by our algorithm during a considerable length of 795 frames or about 90 seconds. Although the sizes of the targets significantly change, no identity switches occur, as can be seen from Fig. 9(a), where the complete trajectories from all trackers are shown in the same image. Furthermore, the tracker robustly handles the significant partial and complete occlusions, which are caused by static objects and other tracking targets, as well as the highly dynamic motion of some targets, which are suddenly stopping, moving backwards, or in circles.

During the entire sequence, our method returns only 4 short false positive trajectories (marked by the red arrows in Fig. 9(a)). They are caused by trackers that are erroneously initialized because of persistent false positive detections in the initialization region at the image borders.

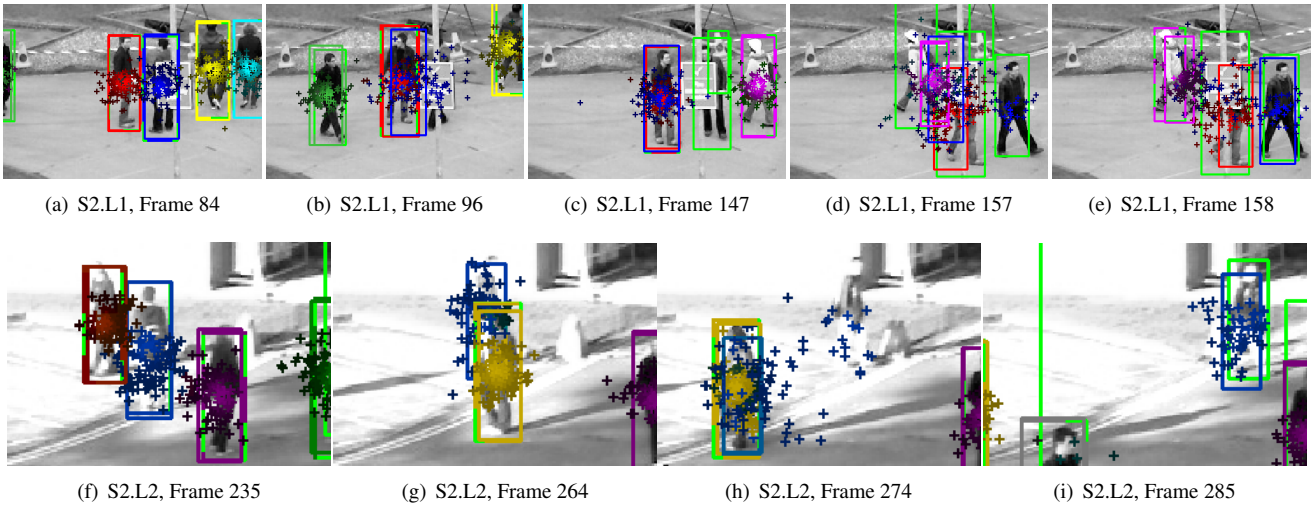
The HOG detection algorithm does not consistently find all pedestrians throughout the sequence (only about 80% of all targets) and regularly produces false positive detections (about 50% of all detections), as can be seen in Figs. 7 and 8). Given the output of the object detector, the average computation time per frame is around 1s.

In Fig. 8 (top row), we show a sequence of frames to illustrate how our algorithm handles situations with severe occlusions. In Fig. 8(a), all trackers are associated with a detection. The target represented by the blue tracker then moves towards the road sign and becomes occluded (see Fig. 8(b)). Since no detection is available, the particles propagate towards nearby areas of high detector confidence density (*i.e.*, to the target of the red tracker). After 50 frames, the target reappears from behind the road sign and is detected again (see Fig. 8(c)). However, the detection is not associated with the blue tracker yet, because the target is still partially occluded and therefore the classifier score is low. The more the target is visible again, the more particles represent the correct target, thanks to the classifier term (see Fig. 8(d)). Finally, all targets are correctly found again in Fig. 8(e).

5.2. PETS Tracking Tasks S2.L2 and S2.L3

For the two tracking tasks S2.L2 and S2.L3, two predetermined targets per sequence have to be tracked. Because our algorithm automatically initializes particle filters on all detected targets (*i.e.*, not just for the targets required for this

[1] For the complete results, please watch the accompanying videos: `www.vision.ee.ethz.ch/~bremicha/tracking/`



(a) S2.L1, Frame 84 (b) S2.L1, Frame 96 (c) S2.L1, Frame 147 (d) S2.L1, Frame 157 (e) S2.L1, Frame 158

(f) S2.L2, Frame 235 (g) S2.L2, Frame 264 (h) S2.L2, Frame 274 (i) S2.L2, Frame 285

Figure 8: Particle output (colored particles and their mean modes), HOG detections (green), associated detections (dashed). *First row:* The object detector often returns false positive detections and does not find all targets. Although the red and blue particle filters temporarily represent the same target (see *(b)*, *(c)*), the tracking algorithm recovers after the occlusion *(e)*. *Second row:* The appearance of the targets changes because of sunlight, and the detector does not always find the targets (*(g)*, *(h)*). Hence, the classifier does not adapt well to the new appearance, causing particle drift *(h)*. However, some particles remain on the correct target *(h)* because of the multi-modality of the particle filter, allowing the tracker to recover *(i)*.

task), we manually select the corresponding result trajectories for the evaluation after running our algorithm.

These sequences S2.L2 (length 436 frames) and S2.L3 (length 240 frames) mainly pose two challenges to our algorithm. First, target appearance changes during the sequences. This is caused by different lighting conditions in the different image areas or when a target turns with respect to the camera position during the sequence (*e.g.*, one target is only visible frontally in the beginning, but turns when walking in a different direction). Second, the people in the crowd move very consistently, regularly occluding each other. Therefore, identity switches are likely to happen. However, our algorithm manages to robustly handle these problems as we demonstrate in the following.

The male target (blue in Figs. 9(b) and 10) is visible for about 350 frames, during which it is heavily occluded sometimes and its appearance changes frequently. Still, our algorithm tracks this person without identity switch. In Fig. 8 (bottom row), the particle output of the algorithm is shown during the most critical phase when entering the well lit area of the image. In this area, the target is only rarely detected by the object detector (see Figs. 8(g) and 8(h)), thus the classifier is not updated and does not adapt very well to the changing appearance. However, because of the multi-modality of the particle filter, some particles remain on the correct target, although the strongest mode is temporarily on another person (Fig. 8(g)). In Fig. 8(i), the tracker recovers again, having the strongest mode of the particle distribution on the correct target.

The female target (red in Figs. 9(b) and 10) leaves the field of view for about 220 frames. When re-entering the scene, the algorithm initializes a new tracker (green, see Figs. 9(b) and 10), causing an identity switch. For the first few frames after the reappearance, this tracker is located on another person before switching to the correct target when it gets fully visible.

To avoid that a new tracker is initialized for a previously observed target that temporarily left the scene, the algorithm would have to deactivate trackers instead of immediately terminating them. Then, it could check for each target entering the field of view whether the same target has been observed before, and it could reactivate the corresponding, already existing tracker. However, this is currently not implemented.

In the sequence S2.L3, our algorithm consistently tracks the two targets (see Figs. 9(c) and 10), although the yellow target is completely occluded by the magenta target for about 170 frames in the beginning of the sequence, as shown in Fig. 7. Although the tracking task gets harder when the targets join the approaching crowd, our algorithm locates the correct targets very precisely most of the time (see Fig. 10). Only when walking behind the road sign in the middle of the field of view, the trackers are temporarily slightly imprecise, but immediately recover when the targets become visible again and are detected.

To sum up, without carefully (but fully automatically) selecting the detections that guide the particle filters, the trackers would be misled by false positive detections or by

detections on other persons. For this purpose, the online trained classifiers are of great help for data association. During an occlusion (*i.e.*, if no matching detection is associated with a tracker) or if the detector fails to detect person (missing detections), the tracker is mainly guided by the detector confidence density term. Finally, if a target is only partially visible, the classifiers help locate the particles precisely.

6. Conclusion

We presented an algorithm for Markovian multi-object tracking-by-detection. The main ideas are to (1) carefully select the final detections using online trained classifiers to handle false positive detections, and to (2) exploit the continuous, intermediate output of state-of-the-art detection algorithms to overcome missing detections. We discussed the different components in detail, demonstrating their importance and limitations.

The target of this work was to demonstrate the capabilities of a tracking-by-detection algorithm that relies only on 2D image information from one single, uncalibrated camera. Our algorithm achieved to successfully solve the PETS'09 tracking tasks with high accuracy and precision. However, the most important vulnerability of our algorithm is its dependency on the detector output. Therefore, the most potential to improve our algorithm is to support the object detector by using a ground plane, camera calibration or a scene model. Of course, also the tracking algorithm itself could benefit from such scene-specific information. Secondly, the input from multiple, synchronized cameras could be used for tracking, which would help especially to resolve occlusion situations. However, both scene specific information as well as multi-camera input is usually not available for arbitrary data sources, limiting the application area of such an extended algorithm.

Acknowledgments: We gratefully acknowledge support by the EU project HERMES (IST-027110).

References

[1] M. Andriluka, S. Roth, and B. Schiele. People-tracking-by-detection and people-detection-by-tracking. In *CVPR*, 2008.

[2] S. Avidan. Ensemble tracking. In *CVPR*, 2005.

[3] J. Berclaz, F. Fleuret, and P. Fua. Robust people tracking with global trajectory optimization. In *CVPR*, 2006.

[4] M. D. Breitenstein, F. Reichlin, B. Leibe, E. Koller-Meier, and L. V. Gool. Robust tracking-by-detection using a detector confidence particle filter. In *ICCV*, 2009.

[5] Y. Cai, N. de Freitas, and J. J. Little. Robust visual tracking for multiple targets. In *ECCV*, 2006.

[6] N. Dalal and B. Triggs. Histograms of oriented gradients for human detection. In *CVPR*, 2005.

[7] T. Fortmann, Y. Bar Shalom, and M. Scheffe. Sonar tracking of multiple targets using joint probabilistic data association. *IEEE J. Oceanic Engineering*, 8(3):173–184, 1983.

[8] J. Giebel, D. Gavrila, and C. Schnörr. A bayesian framework for multi-cue 3d object tracking. In *ECCV*, 2004.

[9] H. Grabner and H. Bischof. On-line boosting and vision. In *CVPR*, 2006.

[10] C. Huang, B. Wu, and R. Nevatia. Robust object tracking by hierarchical association of detection responses. In *ECCV*, 2008.

[11] M. Isard and A. Blake. Condensation—conditional density propagation for visual tracking. *IJCV*, 29(1):5–28, 1998.

[12] S. M. Khan and M. Shah. A multiview approach to tracking people in crowded scenes using a planar homography constraint. In *ECCV*, 2006.

[13] B. Leibe, A. Leonardis, and B. Schiele. Robust object detection with interleaved categorization and segmentation. *IJCV*, 77(1-3):259–289, 2008.

[14] B. Leibe, K. Schindler, and L. Van Gool. Coupled detection and trajectory estimation for multi-object tracking. In *ICCV*, 2007.

[15] K. Okuma, A. Taleghani, N. De Freitas, J. Little, and D. Lowe. A boosted particle filter: Multitarget detection and tracking. In *ECCV*, 2004.

[16] A. G. A. Perera, C. Srinivas, A. Hoogs, G. Brooksby, and W. Hu. Multi-object tracking through simultaneous long occlusions and split-merge conditions. In *CVPR*, 2006.

[17] D. Reid. An algorithm for tracking multiple targets. *IEEE Trans. Automatic Control*, 24(6):843–854, 1979.

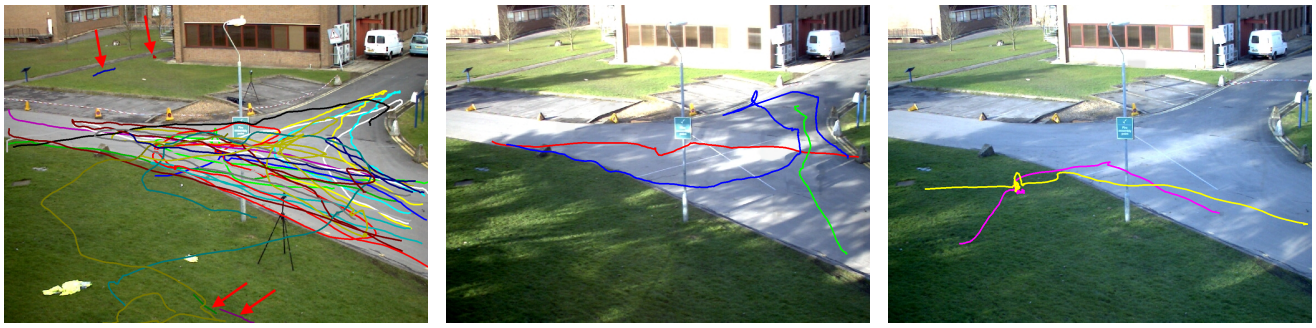
[18] D. Schulz, W. Burgard, D. Fox, and A. B. Cremers. Tracking multiple moving targets with a mobile robot using particle filters and statistical data association. In *ICRA*, 2001.

[19] X. Song, J. Cui, H. Zha, and H. Zhao. Vision-based multiple interacting targets tracking via on-line supervised learning. In *ECCV*, 2008.

[20] C. Stauffer and W. E. L. Grimson. Learning patterns of activity using real-time tracking. *PAMI*, 22(8):747–757, 2000.

[21] J. Vermaak, A. Doucet, and P. Perez. Maintaining multi-modality through mixture tracking. In *ICCV*, 2003.

[22] B. Wu and R. Nevatia. Detection and tracking of multiple, partially occluded humans by bayesian combination of edgelet based part detectors. *IJCV*, 75(2):247–266, 2007.



(a) S2.L1 (b) S2.L2 (c) S2.L3

Figure 9: The resulting trajectories of all targets for the PETS'09 tracking tasks (view 001). In (*a*), the four short false positive trajectories are denoted by the arrows. In (*b*) and (*c*), only the trajectories for the two predetermined targets are shown.

Figure 10: Exemplary tracking results for the PETS'09 tracking datasets: S2.L1 (*top*), S2.L2 (*middle*) and S2.L3 (*bottom*). *Please watch the accompanying videos for the complete results.*