Multi-Person Tracking with Sparse Detection and Continuous Segmentation

Dennis Mitzel[1], Esther Horbert[1], Andreas Ess[2], Bastian Leibe[1]

[1] UMIC Research Centre RWTH Aachen University, Germany
[2] Computer Vision Laboratory, ETH Zurich, Switzerland

Abstract. This paper presents an integrated framework for mobile street-level tracking of multiple persons. In contrast to classic tracking-by-detection approaches, our framework employs an efficient level-set tracker in order to follow individual pedestrians over time. This low-level tracker is initialized and periodically updated by a pedestrian detector and is kept robust through a series of consistency checks. In order to cope with drift and to bridge occlusions, the resulting tracklet outputs are fed to a high-level multi-hypothesis tracker, which performs longer-term data association. This design has the advantage of simplifying short-term data association, resulting in higher-quality tracks that can be maintained even in situations where the pedestrian detector does no longer yield good detections. In addition, it requires the pedestrian detector to be active only part of the time, resulting in computational savings. We quantitatively evaluate our approach on several challenging sequences and show that it achieves state-of-the-art performance.

1 Introduction

In this paper, we address the problem of multi-person tracking with a camera mounted on top of a moving vehicle, *e.g*. a mobile robot. This task is very challenging, since multiple persons may appear or emerge from occlusions at every frame and need to be detected. Since background modeling [1] is no longer applicable in a mobile scenario, this is typically done using visual object detectors [2]. Consequently, tracking-by-detection has become the dominant paradigm for such applications [3–8]. In this framework, a generic person detector is applied to every frame of the input video sequence, and the resulting detections are associated to tracks. This leads to challenging data association problems, since the detections may themselves be noisy, containing false positives and misaligned detection bounding boxes [2]. Several approaches have been proposed to address this issue by optimizing over a larger temporal window using model selection [5], network flow optimization [9], or hierarchical [8] or MCMC data association [10].

Intuitively, this complex data association seems to be at least to some degree an overkill. Once we have detected a person in one frame, we know its appearance and should be able to use this information in order to disambiguate future data associations. This has been attempted by using person-specific color descriptors (*e.g*. [4–6]) or online-trained classifiers [11]. The difficulty here is however that no precise segmentation is given – the detector bounding boxes contain many background pixels and the persons' limbs may undergo considerable articulations, causing the classifiers to drift.

Another problem of tracking systems that only rely on detector input is that they will not work in situations where the detectors themselves fail, *e.g.* when a person gets too close to the camera and is partially occluded by the image borders. [6] explicitly point out those situations as failure cases of their approach.

In this paper, we propose to address those problems by complementing the detection-based tracking framework with a robust image-level tracker based on level-set (LS) segmentation. In this integration, a high-level tracker only initializes new tracklets from object detections, while the frame-to-frame target following and data association is taken over by the image-based tracker. The resulting tracked target locations are then transmitted back to the high-level tracker, where they are integrated into 3D trajectories using physically plausible motion models.

This combination is made possible by the great progress LS segmentation and tracking approaches have made in recent years [12]. Approaches are now available that can obtain robust tracking performance over long and challenging sequences [13]. In addition, LS trackers can be efficiently implemented using narrow-band techniques, since they need to process only a small part of the image around the tracked contour. However, the targeted integration is far from trivial. The LS tracking framework has originally been developed for following individual targets over time and has mostly been evaluated for tasks where a manual initialization is given [12, 13]. Here, we need to automatically initialize a large number of tracklets from potentially inaccurate detections. In addition, we need to deal with overlaps and partial occlusions between multiple followed persons, as well as with tracker drift from changing lighting conditions and poor image contrast. Finally, we need to account for cases where a person gets fully occluded for a certain time and comes into view again a few frames later. In this paper, we show how those challenges can be addressed by a careful interplay of the system components.

Our paper makes the following contributions: (1) We demonstrate how LS trackers can be integrated into a tracking-by-detection framework for robust multi-person tracking. (2) Our approach is based on the idea to track each individual pedestrian by an automatically initialized level-set. We develop robust methods for performing this initialization from object detections and show how additional geometric constraints and consistency checks can be integrated into the image-based LS tracker. (3) The tracked person contours in each video frame are automatically converted to 3D world coordinates and are transmitted to the high-level tracker, which integrates the position evidence into a robust multi-hypothesis trajectory estimation approach making use of physical motion models. This high-level tracker is responsible for initializing new tracks, correcting the low-level tracker's predictions when drift occurs, and tracking person identities through occlusions. (4) We experimentally demonstrate that this proposed integration achieves robust multi-person tracking performance in challenging mobile scenarios. In particular, as our approach does not depend on continuous pedestrian detection, it can also continue tracking persons that are only partially visible. (5) An interesting property of our integration is that it does not require the object detector to be executed for every video frame. This is especially relevant for the deployment on mobile platforms, where real-time performance is crucial and computational resources are notoriously limited. We experimentally investigate at what intervals object detections are still required for robust system-level performance.

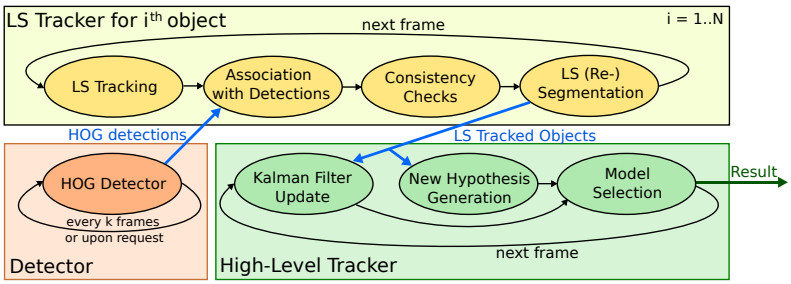


Fig. 1. System-level view of our proposed end-to-end tracking framework.

The following section discusses related work. After that, Sec. 2 presents our proposed end-to-end tracking framework. Sec. 3 introduces the basic algorithmic components for LS tracking and trajectory estimation. Sec. 4 then describes the details of the integration, and Sec. 5 presents experimental results.

Related Work. Multi-object tracking from a mobile platform is a core capability for many applications in mobile robotics and autonomous vehicles [14]. While early approaches have been developed for aerial scenarios [15, 16], an application on ground-level poses significant additional difficulties. Robust multi-person tracking in such challenging situations has only recently become feasible by the development of powerful tracking-by-detection approaches [4, 7, 14, 5, 6]. Various strategies have been developed for solving the challenging data association problems encountered here. However, most of them regard only a single-layer tracker [11, 3, 5–7], which sometimes makes the problem unnecessarily hard. Most directly related to our approach are the multi-layer models of [16, 15], which also initialize a number of low-level trackers to follow individual objects and later integrate their results in a high-level tracker. However, their frameworks are based on aerial scenarios, where adaptive background modeling is still feasible. [8] also propose a hierarchical data association framework that links detection responses to form tracklets at an image level, before fusing the tracklets and integrating scene constraints at higher levels. Their approach is however targeted at a surveillance application with a static camera. [17] integrates multiple short and low-confidence tracklet hypotheses into consistent tracks using MCMC. In contrast, our approach creates long and highly confident tracklets for individual persons under specific conditions of an LS tracker and integrates them into an EKF-based multiple-hypothesis tracker. To our knowledge, ours is the first approach that integrates segmentation-based LS-trackers [13, 12] with a tracking-by-detection framework for street-level mobile tracking.

2 Integrated Tracking Framework

Fig. 1 shows a system-level overview of our proposed integrated tracking framework. The system is initialized by detections from a pedestrian detector. For each detected person, an independent LS tracker (a *tracklet*) is initialized, which follows this person's motion in the image space. The LS tracker is kept robust through a series of consistency checks and transmits the tracked person's bounding box to the high-level tracker after every frame. The high-level tracker in turn converts the bounding boxes to ground plane

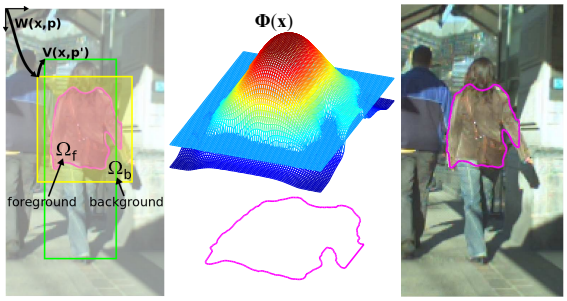


Fig. 2. Level-set segmentation. The contour separates object $\mathbf{\Omega}_f$ from background $\mathbf{\Omega}_b$ in a reference frame given by the warp $W(\mathbf{x}, \mathbf{p})$, which is related to the person's bounding box by the displacement $V(\mathbf{x}, \mathbf{p}')$. This contour is the zero level-set of the embedding function $\mathbf{\Phi}$.

coordinates and integrates them into physically plausible trajectories using the model selection framework described in Sec. 3.2. During regular operation, the object detector only needs to be activated in regular intervals in order to prevent existing tracklets from degenerating and to start new ones for newly appearing pedestrians. In addition, tracklets can request new detections when they become uncertain. Overall, this results in considerable computational savings, as we will show in Sec. 5.

Setup. Similar to previous work on mobile pedestrian tracking [14, 5, 6], we assume a setup of a stereo camera rig mounted on a mobile platform. From this setup, we obtain structure-from-motion (SfM), stereo depth, and a ground plane estimate for every frame. All subsequent processing is then performed only on the left camera stream.

3 Algorithmic Components

3.1 Level-Set Tracking

Like [13], we use a probabilistic level-set framework, which first performs a segmentation and in the next frames a rigid registration and shape adaptation. The object shape is defined by the zero level-set of an embedding function $\mathbf{\Phi}(\mathbf{x})$ (Fig. 2) acting on pixel locations $\mathbf{x}$ with appearance $\mathbf{y}$. This level-set is evolved in order to maximize the accordance with learned foreground and background appearance models M_f and M_b, while fulfilling certain constraints on the shape of the embedding function and of the contour.

Segmentation. The variational level-set formulation for the segmentation consists of three terms which penalize the deviation from the foreground and background model, the deviation of the embedding function from a signed distance function [18], and the length of the contour. A segmentation is achieved by optimizing this energy functional with the following gradient flow [13]:

$$\frac{\partial P(\mathbf{\Phi}, \mathbf{p}|\mathbf{\Omega})}{\partial \mathbf{\Phi}} = \underbrace{\frac{\delta_\epsilon(\mathbf{\Phi})(P_f - P_b)}{P(\mathbf{x}|\mathbf{\Phi}, \mathbf{p}, \mathbf{y})}}_{\text{deviation from fg/bg model}} - \underbrace{\frac{1}{\sigma^2}\left[\nabla^2\mathbf{\Phi} - div(\frac{\nabla\mathbf{\Phi}}{|\nabla\mathbf{\Phi}|})\right]}_{\text{deviation from signed dist. fct.}} + \underbrace{\lambda\delta_\epsilon(\mathbf{\Phi})div(\frac{\nabla\mathbf{\Phi}}{|\nabla\mathbf{\Phi}|})}_{\text{length of contour}} \tag{1}$$

where $P(\mathbf{x}_i|\mathbf{\Phi},\mathbf{p},\mathbf{y}_i) = H_\epsilon(\mathbf{\Phi}(\mathbf{x}_i))P_f + (1 - H_\epsilon(\mathbf{\Phi}(\mathbf{x}_i)))P_b$, ∇^2 is the Laplacian, H_ϵ is a smoothed Heaviside step function and δ_ϵ its derivative, a smoothed Dirac delta function. $\mathbf{\Omega} = \{\mathbf{\Omega}_f, \mathbf{\Omega}_b\}$ denotes the foreground/background pixels in the object frame.

P_f and P_b are the pixel-wise posteriors of the foreground and background models given the pixel appearance. Those models are obtained from the pixels inside and outside the contour during the first segmentation. The segmentation is performed in several iterations and the models are rebuilt in every iteration. In the subsequent tracking steps, the model parameters M_f and M_b are only slightly adapted to the current image in order to achieve higher robustness.

Tracking. Similar to image alignment, the tracking part aims at warping the next frame such that its content best fits the current level-set. This way, the location of the tracked object is obtained. The warp $W(\mathbf{x},\mathbf{p})$ is a transformation of the reference frame with parameters $\mathbf{p}$. Any transformation forming a group can be used here, *e.g.* affine transformations. In our application for pedestrian tracking, we currently use only translation+scale. For optimizing the location, the next image is incrementally warped with $\Delta\mathbf{p}$ until convergence [13]:

$$\Delta\mathbf{p} = \left[\sum_{i=1}^{N} \frac{1}{2P(\mathbf{x}_i|\mathbf{\Phi},\mathbf{p},\mathbf{y}_i)} \left[\frac{P_f}{H_\epsilon(\mathbf{\Phi}(\mathbf{x}_i))} - \frac{P_b}{(1-H_\epsilon(\mathbf{\Phi}(\mathbf{x}_i)))}\right] \mathbf{J}^\mathsf{T}\mathbf{J}\right]^{-1} \times \sum_{i=1}^{N} \frac{(P_f - P_b)\mathbf{J}^\mathsf{T}}{P(\mathbf{x}_i|\mathbf{\Phi},\mathbf{p},\mathbf{y}_i)} \quad (2)$$

with $\mathbf{J} = \delta_\epsilon(\mathbf{\Phi}(\mathbf{x}_i))\nabla\mathbf{\Phi}(\mathbf{x}_i)\frac{\partial W}{\partial \Delta\mathbf{p}}$, where $\frac{\partial W}{\partial \Delta\mathbf{p}}$ is the Jacobian of the warp.

Appearance Models. [13] only uses color for the foreground and background model. We found that in our application, this yields rather unreliable segmentations for pedestrians, since other people or background structures often contain similar colors. We therefore extend the approach by also including stereo depth information.

For segmentation, we use the median depth of the foreground area. Unlike the color distribution, the median depth will not stay the same during the following frames. For tracking, we therefore use a simple motion model which computes an expected distance range for each pedestrian according to the last median depth and a maximum velocity. Each depth value in the image is then assigned a probability according to a Gaussian distribution around the median depth or the expected depth, respectively. The color models are represented as L*a*b histograms with 32^3 bins. The two probabilities for color and depth are individually normalized as in [13] and then merged with a weighting factor α (set to 0.1 in all of our experiments).

$$P_i = (1-\alpha)P_{i,color} + \alpha P_{i,depth}, \quad i \in \{f, b\}\ , \quad (3)$$

3.2 Tracking-by-Detection

For the high-level tracker, we use a simplified version of the robust multi-hypothesis tracking framework by [5]. We first describe the basic approach, as it would be applied for pure tracking-by-detection. Section 4 then details how this approach is adapted through the integration with the level-set tracker.

In brief, the approach works as follows. Detected pedestrian locations are converted to 3D world coordinates using the current camera position from SfM together with an

(a) (b) (c) (d) (e) (f)

Fig. 3. Initialization of the LS tracker: (a) detection box (green), initial object frame (yellow), and initialization of the level-set (magenta); (b,c) evolved level-set after 40 and 150 iterations; (d) level-set transferred to next frame; (e) after warping; (f) after shape adaptation (5 iterations).

estimate of the scene's ground plane. These measurements are collected in a spacetime volume, where they are integrated into multiple competing trajectory hypotheses. The final hypothesis set is then obtained by applying model selection in every frame.

Trajectory Estimation. We model pedestrian trajectories by Kalman filters with a constant-velocity motion model on the ground plane, similar to [6]. When new observations become available in each frame, we first try to extend existing trajectory hypotheses by the new evidence. In addition, we start a new trajectory hypothesis from each new detection and try to grow it by applying a Kalman filter backwards in time through the spacetime volume of past observations. This step allows us to recover lost tracks and bridge occlusions. As a consequence of this procedure, each detection may end up in several competing trajectory hypotheses.

Model Selection. For each frame, we try to find the subset of trajectory hypotheses that provides the best explanation for the collected observations. This is done by performing model selection in a Minimum Description Length framework, as in [5]. A trajectory's score takes into account the likelihood of the assigned detections under its motion and appearance model (represented as a color histogram). Trajectory hypotheses interact through penalties if they compete for the same detections or if their spacetime footprints overlap. For details of the mathematical formulation we refer to [5].

Assigning Person Identities. As the model selection procedure may choose a different hypothesis set in each frame, a final step is required in order to assign consistent person IDs to the selected trajectories. This is done by maintaining a list of active tracks and assigning trajectories to them based on the overlap of their supporting observations.

4 Combined Tracker

We now present the stages of our combined tracking framework. The difficulty of the street-level mobile tracking task brings with it a number of non-trivial challenges, which we address by consistency checks and carefully modeled interactions between the components of the tracking framework.

Object Detection. For pedestrian detection, we apply the widely used HOG detector [19] in the efficient *fastHOG* GPU implementation by [20]. Detections inconsistent with the scene geometry are filtered out by enforcing a ground plane corridor.

(a) (b) (c) (d) (e) (f)

Fig. 4. Adaptation to lighting changes: (a-c) tracked shape becomes too small due to lighting changes; (d,e) level-set re-initialization is triggered; (f) tracking can continue.

Level-Set Initialization. Upon initialization, the LS tracker tries to segment the torso of the person inside a detection box. To this end, a new level-set embedding function is initialized with a rectangular box (see Fig. 3), and the level-set segmentation is iterated for 150 steps. In the next frame, the contour is tracked and the resulting warp is applied to the object frame and the associated detection box in order to obtain the new object position. Afterwards, the level-set shape is adapted for 5 iterations. We track only the person's torso, since this body part deforms only very little, requiring fewer shape adaptation iterations than tracking the full body. This speeds up level-set tracking and increases the robustness, since it limits the amount of "bleeding" that can occur to similar-colored background pixels. To infer the person's full extent, we maintain the transformation $V(\mathbf{x}, \mathbf{p}')$ from the warped reference frame to the original bounding box.

Multi-Region Handling and Overlap Detection. When tracking several persons, each of the tracked contours is represented by its own level-set. Even if there are overlaps, the level-sets will not interact directly (as, *e.g.*, in [21]). Instead, we use the stereo depth in order to resolve overlaps. All tracked persons are sorted according to their distance from the camera and the closest person is updated first. All pixels belonging to the resulting segmentation are masked out, such that they cannot be used by the remaining persons.

This leaves us with some persons that are only partially visible, which is in fact the same case as a person leaving the image frame. We developed a method for dealing with partial visibility without losing shape information. As can be seen in eq. (2), only a narrow band of pixels around the contour, which is determined by $\delta_\epsilon(\mathbf{\Phi})$, is taken into account for tracking. If pixels are masked out or are outside the image frame, we set δ_ϵ to zero for those pixels, which will result in tracking only the visible part of the contour. Thus, if an object becomes completely visible again, the shape will still fit. Objects are discarded if only a small part of the area inside the contour (50% for person-person occlusions, 20% for occlusions by image borders) remains visible.

Level-Set Re-initialization. Lighting changes or similar colors near the object can cause the contour to shrink during tracking (see Fig. 4) or to bleed out during shape adaptation. By periodically updating a tracklet bounding box with new detector bounding boxes, it is possible to identify degenerating shapes based on their size in relation to the bounding box. This is done by first performing the level-set tracking step for adapting the contour to the new image and then matching the tracked location to new detector boxes. If the box overlap (measured by intersection-over-union) is above 0.5,

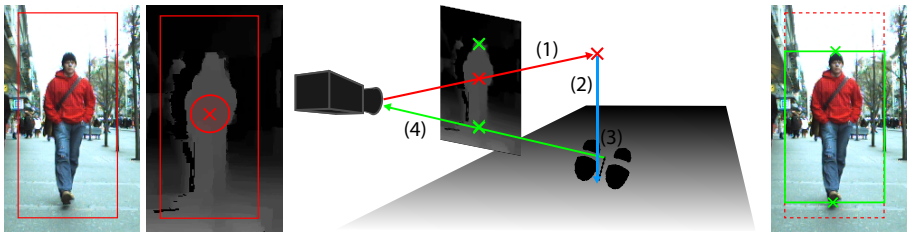


Fig. 5. Depth-based bounding box correction: (a) original bounding box; (b) depth map; (c) correction procedure; (d) corrected bounding box (see text for details).

the detection box is used to update the relationship $V(\mathbf{x}, \mathbf{p}')$ between box and warp. The level-set contour itself is only updated if its area gets too small or too large with respect to the updated box, or if 20% of its content lie outside the box. Thus, the tracklet integrity is maintained and an ID change is avoided (*c.f.* Fig. 4).

Consistency Checks. For robust operation, it is necessary to check the consistency of the tracking results. An object could be occluded, leave the image frame or be lost for other reasons. This may not even have any effect on the convergence of the LS tracker, which might get stuck on some local image structure, resulting in a wrong track. We therefore perform the following checks in order to identify corrupted tracklets. (1) If the object is occluded and only background colors remain, the shape will typically shrink massively within a few frames. If such a case is detected, the tracklet is terminated. (2) We keep track of the median depth inside the tracked contour and react if this value changes too fast. We distinguish two cases here: If the median depth decreases too fast, this indicates an occlusion by another object; if the depth increases too fast, the object was probably lost. We terminate the tracklets in both cases. (3) Finally, objects whose median depth does not fit their ground-plane distance are also discarded. Typically, a failed consistency check indicates a tracking failure and will result in a request for the detector to be activated in the next frame. An exception are cases where an occlusion is "explained" by the high-level tracker (see below), or when the object is close to the image boundary and is about to leave the image.

Depth-based Bounding Box Correction. Level-set (re-)initialization and high-level 3D trajectory integration require accurately aligned bounding boxes. In general, the HOG detector however yields detections with a certain border area. Similarly, the boxes provided by the LS tracker may drift due to articulations and shape changes of the level-set contour and need to be corrected. We therefore apply the following correction procedure both to new detections and after each level-set tracking step. Starting from the original bounding box (Fig. 5(a)), we first compute the median depth around the bounding box center (Fig. 5(b)). We then determine the corresponding 3D point using the camera calibration from SfM and project it onto the ground plane (Fig. 5(c), steps(1)+(2)). We add a fixed offset in the viewing direction in order to determine the person's central foot point, and finally project the resulting 3D point back to the image (Fig. 5(c), steps (3)+(4)). This determines the bottom line of the corrected bounding box. The top line is found by searching for the highest point inside the bounding box that is within 0.5m of the median depth (Fig. 5(d)). As a final step, we verify that the re-

sulting bounding box aspect ratio is in the range $[\frac{1}{3}, \frac{2}{3}]$. Bounding boxes falling outside this range are rejected as likely false positives.

Requesting New Detections. New detections are requested in the following cases: (1) if a tracklet has not received an updated detection in the last k frames; (2) if a tracking failure cannot be explained by an occlusion or by the tracked person leaving the image; (3) if no request has been issued for k frames (*e.g.*, since no object is currently tracked). A tracklet will not request new detections if it is close to the image boundary, as the chance for finding a detection there would be small. If a tracklet receives no updated detection despite its request, it will repeat the request, but will continue to be tracked as long as it passes the consistency and depth correction checks.

Integration with High-Level Tracker. The high-level tracker's task is to integrate the tracklet bounding boxes into physically plausible 3D trajectories. This is done by first creating an *observation* at each tracked person's 3D foot point and then associating this observation to trajectory hypotheses. The overall procedure is similar to the general tracking-by-detection framework described in Sec. 3. However, we make the following changes in order to account for the additional information provided by the LS tracker.

Since we already know the tracklet identity of each observation from the LS tracker, we can use this information in order to simplify data association. Thus, we first try to extend each existing trajectory hypothesis by searching for an observation matching the trajectory's currently followed tracklet ID in a gating area around the Kalman filter prediction. If such an observation can be found, it will directly be associated with the trajectory. Note that in this case, only the motion model is considered; the appearance is assumed to be correct due to the association performed by the LS tracker. In case no observation with the correct tracklet ID can be found, we try to find the best-matching observation under the trajectory's motion and appearance model (again within a gating area determined by the Kalman filter uncertainty). If such a new observation can be found, the trajectory takes on the new tracklet ID, thus connecting the two tracklets. This latter case can occur if the LS tracker diverges and fails the consistency checks (in which case the tracklet will be terminated), if the tracked bounding box is rejected by the depth correction (in which case the tracklet may persist for up to k frames and can be recovered), or if the tracked object is occluded or leaves the image.

In addition to the above, each observation is used to start a new trajectory hypothesis, which searches backwards in time in order to find a potentially better explanation for the observed data. This makes it possible to automatically create tracks for newly appearing persons and to correct earlier tracking errors. The final set of accepted tracks is then obtained by performing model selection, as described in Section 3.2.

Tracking through Occlusions. As motivated above, a main advantage of the image-based low-level tracker, compared to a pure tracking-by-detection approach, is that it simplifies data association, thus making it easier to integrate observed pedestrian locations into valid tracks. The image-based tracklet generation will however fail when the tracked person gets occluded, which often occurs in practice. This is a limitation of any image-based tracking approach. While strategies can be devised to cope with short-term occlusions at the image-level, they would make this component unnecessarily complex. In our approach, we instead address this issue by explicit occlusion handling on the

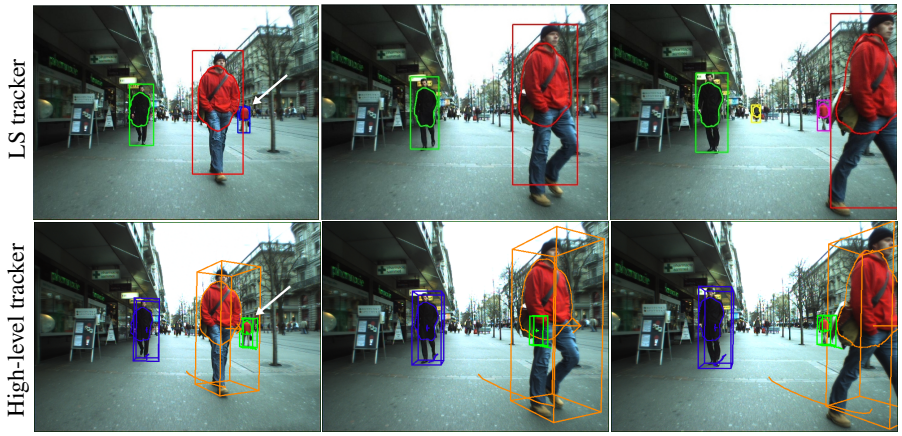


Fig. 6. Example for the occlusion handling process: (top row) contours tracked by the LS tracker; (bottom row) output of the high-level tracker. When the distant person is temporarily occluded, its LS tracklet is terminated. As soon as the occlusion is over, a new tracklet is started. The high-level tracker connects both tracklets through the occlusion and maintains the person's identity.

high-level tracker's side. In order to bridge short-time occlusions, we keep potentially occluded trajectories alive for up to 15 frames and extrapolate their position on the ground plane using the Kalman filter. Since the latter's positional uncertainty grows with the absence of observations, the corresponding person can likely be associated to the predicted trajectory again when reappearing from the occlusion.

In addition, the high-level tracker can predict person-person occlusions and reinitialize the image-based tracker when those are over. For this, we backproject the predicted 3D bounding box of each tracked person into the image and compute the bounding box overlap using the intersection-over-union criterion. If the overlap is larger than 0.5, then an occlusion is likely to occur. This information is stored together with the occluded trajectory and is transmitted to the corresponding LS tracklet, which will typically be terminated 1-2 frames later when the consistency check fails. When the corresponding object is predicted to become visible again a few frames later, the object detector is fired in order to recover the person with as little delay as possible. This "safe termination" and subsequent new tracklet generation strategy proved to be robust in our experiments. It is similar in spirit to the *track-suspend-fail* strategy proposed in [15], but our approach extends the idea through the integration of the robust multi-hypothesis tracking framework.

Fig. 6 shows an example where this occlusion handling process is used in practice. Cued by the occlusion prediction and the failed depth consistency check, the LS tracklet is terminated in order to avoid degeneracies (which would be likely in this case due to the similar color distributions). On the high-level tracker's side, the trajectory is however extrapolated through the occlusion. As soon as the occluded person becomes visible again, the object detector is fired again in order to initialize a new LS tracklet, which is correctly associated to the trajectory, maintaining the person's identity.

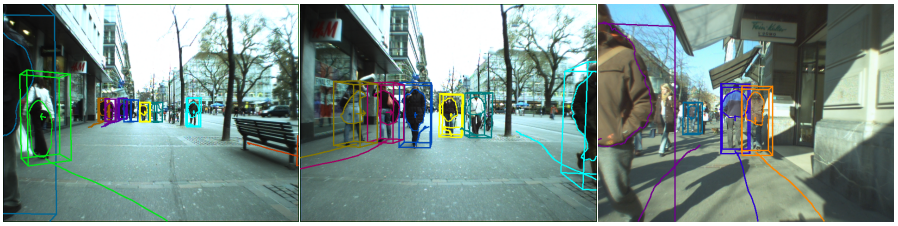


Fig. 7. Examples demonstrating our approach's capability to continue tracking persons close to the camera and/or the image borders, where object detection is no longer applicable.

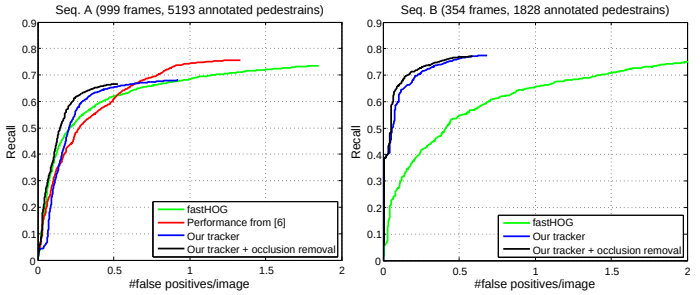
5 Experimental Results

Datasets. We evaluated our approach on two challenging sequences from the Zurich Mobile Pedestrian corpus generously provided by the authors of [6]. We used the sequences BAHNHOF (in the following: "Seq. A") and SUNNY DAY ("Seq. B"). Both sequences were captured with a stereo rig (13-14fps, 640x480). Seq. A (999 frames, with 5193 annotated pedestrians of $\geq$ 60 pixels height) was taken on a crowded sidewalk on a clouded day. Seq. B (999 frames, 354 of which are annotated with 1867 annotations) was captured on a sunny day and contains strong illumination changes. Both sequences come with stereo depth maps, structure-from-motion localization, and ground plane estimates. Similar to [6], we upscale all images to twice their original resolution in order to detect also pedestrians at larger distances. Using the upscaled images, fastHOG performed at 2-3fps (10fps for original images). In contrast to [6, 5], we however only use the left camera stream for detection and tracking, thus reducing the necessary processing effort. All system parameters were kept the same throughout both sequences.

Tracking Performance. Figure 7 shows qualitative results of our approach, demonstrating its capability to continue tracking persons that appear close to the camera or that are partially occluded by the image boundaries. This is a fundamental advantage our tracking framework can offer over pure tracking-by-detection approaches.

In order to assess our approach's performance quantitatively, we adopt the evaluation criteria from [6] and measure the intersection-over-union of tracked person bounding boxes and annotations in every frame. We accept detections having an overlap greater than 0.5 as correct and report the results in terms of *recall vs. false positives per image* (fppi). Fig. 8 shows the resulting performance curves when we set the maximum re-initialization interval to $k = 5$ frames (in blue), together with the baseline of fastHOG (in green). As can be seen, our approach achieves good performance, reaching 65% and 76% recall at 0.5 fppi for Seq. A and Seq. B, respectively. As the bounding box criterion penalizes the tracker's property of predicting a person's location through occlusions (since those cases are not annotated in the test data), we additionally provide the performance curve when filtering out tracked bounding boxes which are more than 50% occluded by other boxes (in black). This results in an additional improvement.

For comparison, we also provide the performance curve reported by [6] on Seq. A, which is also based on HOG detections (shown in red, no such curve is available for Seq. B). This approach integrates detections from both camera streams and thus obtains



	Seq. A	Seq. B
# persons	67	28
mostly hit	54	23
part. tracked	4	1
mostly missed	9	4
false alarms	9	6
ID sw.	6	3
ID sw. (long occ.)	12	3

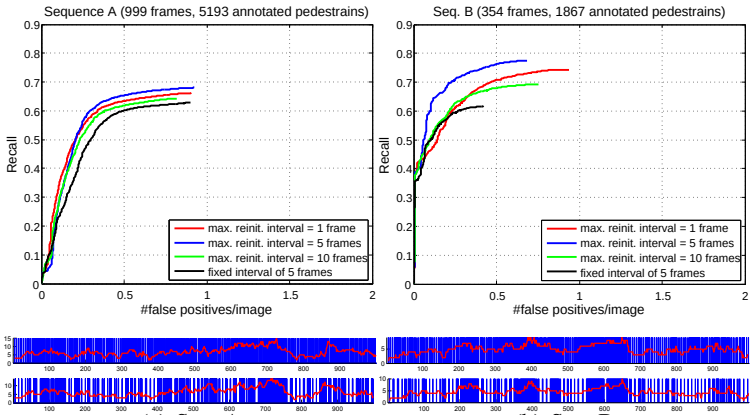
Fig. 8. (left) Quantitative tracking performance of our approach compared to different baselines. (right) Track-level evaluation according to the criteria by [7].

Fig. 9. Example tracking results of our approach on several challenging test sequences.

a higher recall. Its performance should be compared to our blue curve, since no occlusion removal was performed in [6]. Still, it can be seen that our approach achieves better performance in the high-precision range, despite only using a single camera stream. This is a result of the better data association provided by the image-level tracklets.

Fig. 8 (right) also reports a track-level evaluation according to the criteria by [7], showing that most pedestrians are correctly tracked and only few ID switches occur. Fig. 9 shows results of our combined tracker for both test sequences and visualizes the obtained level-set contours. The corresponding result videos are provided on `www.mmp.rwth-aachen.de/projects/eccv2010`. Our system is able to track most of the visible pedestrians correctly in a very busy environment with many occlusions.

Efficiency Considerations. One of our goals was to reduce the dependence on the costly detector. Even though efficient GPU implementations are now available for HOG (*e.g.* [20]), the framerate is still not sufficient for real-time operation in a pure tracking-by-detection context. In addition, the excessive power consumption of GPUs is a major restriction for their use in mobile robotics applications. In contrast, the level-set tracking approach employed here can be very efficiently implemented on regular CPUs. [13] report a framerate of 85Hz for tracking a single target of size 180×180 pixels in their



(a) Seq. A (b) Seq. B

Fig. 10. (top) Tracking performance for the two test sequences when varying the maximum re-initialization interval; (bottom) Frequency of detector activations for both sequences for an interval of 5 (first) and 10 (second) frames. The red curve shows the number of tracked pedestrians.

implementation. In our application, we track targets at a lower resolution of 80×100 pixels and therefore expect even faster performance once our code is fully optimized.

An important consideration in this respect is how often the pedestrian detector needs to be activated for robust tracking performance. Our approach lets the LS tracker request detections whenever required, but enforces a maximum re-initialization interval of k frames. Fig. 10 shows the effective frequency of detector activations when setting this interval to $k \in \{1, 5, 10\}$, together with the resulting tracking performance. A setting of $k = 5$ provides the best tracking quality with a detector activation on average every 1.66 frames. By increasing the maximum interval to 10 frames, the detector activation rate falls to every 2.71 frames at a small loss in recall that is still comparable to [6] at 0.5 fppi. Considering that [6] performed detection in both camera streams, our approach thus requires 5.42 times less detector activations. Finally, we show the performance when activating the detector at a fixed interval of 5 frames, without additional requests. This results in a small drop in recall, but still yields good overall performance.

6 Conclusion

We have presented an integrated framework for mobile street-level multi-person tracking. Our approach combines the advantages of a fast segmentation-based tracker for following individual persons with the robustness of a high-level multi-hypothesis tracking framework for performing longer-term data association. As our experiments have shown, the approach reaches state-of-the-art performance, while requiring fewer detector evaluations than conventional tracking-by-detection approaches. Our results open several interesting research perspectives. The requested detector activations for tracklet re-initialization could be restricted to the tracklet's immediate neighborhood, thus resulting in further speedups. In addition, the obtained level-set segmentation could be a possible starting point for articulated tracking that we plan to explore in future work.

Acknowledgments This project has been funded, in parts, by the EU project EUROPA (ICT-2008-231888) and the cluster of excellence UMIC (DFG EXC 89). We thank C. Bibby and I. Reid for valuable comments for the level-set tracking and for making their evaluation data available.

References

1. Stauffer, C., Grimson, W.: Adaptive Background Mixture Models for Realtime Tracking. In: CVPR'99. (1999)
2. Dollar, P., Wojek, C., Schiele, B., Perona, P.: Pedestrian Detection: A Benchmark. In: CVPR'09. (2009)
3. Okuma, K., Taleghani, A., de Freitas, N., Little, J., Lowe, D.: A Boosted Particle Filter: Multitarget Detection and Tracking. In: ECCV'04. (2004)
4. Andriluka, M., Roth, S., Schiele, B.: People Tracking-by-Detection and People Detection-by-Tracking. In: CVPR'08. (2008)
5. Leibe, B., Schindler, K., Van Gool, L.: Coupled Object Detection and Tracking from Static Cameras and Moving Vehicles. PAMI **30** (2008) 1683–1698
6. Ess, A., Leibe, B., Schindler, K., Van Gool, L.: Robust Multi-Person Tracking from a Mobile Platform. PAMI **31** (2009) 1831–1846
7. Wu, B., Nevatia, R.: Detection and Tracking of Multiple, Partially Occluded Humans by Bayesian Combination of Edgelet Part Detectors. IJCV **75** (2007) 247–266
8. Huang, C., Wu, B., Nevatia, R.: Robust Object Tracking by Hierarchical Association of Detection Responses. In: ECCV'08. (2008)
9. Zhang, L., Nevatia, R.: Global Data Association for Multi-Object Tracking Using Network Flows. In: ECCV'08. (2008)
10. Zhao, T., Nevatia, R., Wu, B.: Segmentation and Tracking of Multiple Humans in Crowded Environments. PAMI **30** (2008) 1198–1211
11. Grabner, H., Grabner, M., Bischof, H.: Real-Time Tracking via On-line Boosting. In: BMVC'06. (2006)
12. Cremers, D., Rousson, M., Deriche, R.: A Review of Statistical Approaches to Level Set Segmentation Integrating Color, Texture, Motion and Shape. IJCV **72** (2007) 195–215
13. Bibby, C., Reid, I.: Robust Real-Time Visual Tracking using Pixel-Wise Posteriors. In: ECCV'08. (2008)
14. Gavrila, D., Munder, S.: Multi-Cue Pedestrian Detection and Tracking from a Moving Vehicle. IJCV **73** (2007) 41–59
15. Kaucic, R., Perera, A., Brooksby, G., Kaufhold, J., Hoogs, A.: A Unified Framework for Tracking through Occlusions and Across Sensor Gaps. In: CVPR'05. (2005)
16. Tao, H., Sawhney, H., Kumar, R.: Object Tracking with Bayesian Estimation of Dynamic Layer Representations. PAMI **24** (2002) 75–89
17. Ge, W., Collins, R.: Multi-target Data Association by Tracklets with Unsupervised Parameter Estimation. In: BMVC'08. (2008)
18. Li, C., Xu, C., Gui, C., Fox, M.: Level Set Evolution without Re-initialization: A New Variational Formulation. In: CVPR'05. (2005)
19. Dalal, N., Triggs, B.: Histograms of oriented gradients for human detection. In: CVPR'05. (2005)
20. Prisacariu, V., Reid, I.: fastHOG – a Real-Time GPU Implementation of HOG. Technical Report 2310/09, Dept. of Engineering Science, University of Oxford (2009)
21. Brox, T., Weickert, J.: Level Set Based Image Segmentation with Multiple Regions. In: DAGM'04. (2004)