Real-Time Multi-Person Tracking with Detector Assisted Structure Propagation

Dennis Mitzel and Bastian Leibe
UMIC Research Centre
RWTH Aachen University, Germany
{mitzel,leibe}@umic.rwth-aachen.de

Abstract

Classical tracking-by-detection approaches require a robust object detector that needs to be executed in each frame. However the detector is typically the most computationally expensive component, especially if more than one object class needs to be detected. In this paper we investigate how the usage of the object detector can be reduced by using stereo range data for following detected objects over time. To this end we propose a hybrid tracking framework consisting of a stereo based ICP (Iterative Closest Point) tracker and a high-level multi-hypothesis tracker. Initiated by a detector response, the ICP tracker follows individual pedestrians over time using just the raw depth information. Its output is then fed into the high-level tracker that is responsible for solving long-term data association and occlusion handling. In addition, we propose to constrain the detector to run only on some small regions of interest (ROIs) that are extracted from a 3D depth based occupancy map of the scene. The ROIs are tracked over time and only newly appearing ROIs are evaluated by the detector. We present experiments on real stereo sequences recorded from a moving camera setup in urban scenarios and show that our proposed approach achieves state of the art performance.

1. Introduction

Robust multi-person tracking is an important prerequisite for the use of mobile service robots in busy urban settings. In this paper, we address the problem of stereo vision based multi-person tracking from a mobile platform with reduced object detector evaluations. Unlike applications with stationary cameras, a mobile setup requires a visual object detection component, since background subtraction is no longer applicable. Following the enormous progress in object detection [7, 11], many robust tracking-by-detection approaches have recently been proposed for this purpose [21, 1, 17, 9, 25, 15, 12]. However, they typically require to evaluate a computationally expensive object detector in each frame, making it hard to achieve real-time performance at the system level.

Two main strategies have been proposed in order to alleviate this problem. The first is to constrain object detection to small image regions-of-interest (ROIs) extracted, *e.g.*, using stereo depth information [12, 14, 4, 3]. These ROIs are generated in each frame and are evaluated by the detector to feed a tracking-by-detection process. Most similar to our approach, Bansal *et al.* [4] extract ROIs from an image by projecting the 3D points from a stereo depth map onto the estimated 2D ground plane. The local maxima of this projection are backprojected to the image, forming the ROIs which are evaluated in each frame by the detector. The detector output is then associated to trajectories using a correlation tracker. As only a small number of ROIs are processed in each frame, their approach nearly reaches real-time performance.

A second strategy is to combine a complex high-level tracker with a cheap low-level tracker that takes the brunt of the work of following individual persons over time [19]. This strategy is based on the idea that once a person is de-

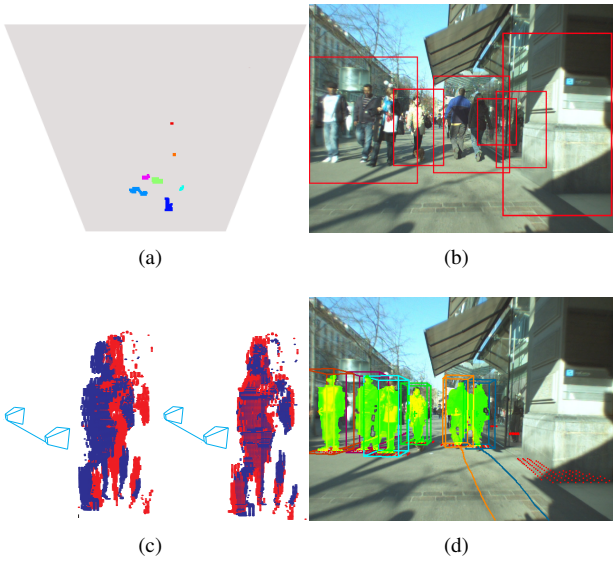


Figure 1: (a) Given stereo range data, we first extract and track ROI candidates from the depth map (top down view). (b) Extracted ROIs in 3D are backprojected to the image. (c) Model (red points) and data (blue points) point clouds are aligned using ICP. (d) Results after high-level tracker association using the ICP tracker

tected in a frame, its appearance will change only slightly in the following frames, making simple low-level tracking feasible. Mitzel *et al*. [19] propose such a hybrid framework based on a cheap level-set tracker and a complex multi-hypothesis high-level tracker. In their framework, the detector is only activated every k frames, (re-)initializing a set of low-level trackers that generate tracklets. The high-level tracker then associates the resulting tracklets to plausible trajectories, taking care of long-term data association and maintaining person identities in case of occlusions. Through this combination fewer detector evaluations are required from the detector than in conventional tracking-by-detection approaches. The approach however suffers from the fact that it may take several frames until newly appearing persons are picked up by the tracker.

In this paper, we explore a combination of those two strategies that combines their advantages. Similar to [4], we extract ROIs based on stereo range data. In contrast to their approach, we however do not simply apply the detector to all ROIs in every frame, but instead track the ROIs over time using cheap Kalman filter based data association and only evaluate newly appearing ROIs with the detector. Similar to [19], we then use verified detections to initialize low-level tracker, which is responsible for frame-to-frame object following. In contrast to them, we however propose to use an ICP based low-level tracker that achieves better localization accuracy making use of the same depth information that was used for generating the ROIs (Fig.1). The output of the ICP tracker is then passed on to a high-level tracker performing long-term data association. As our results will demonstrate, this combination reduces the number of the detector activations significantly, while still maintaining high tracking accuracy and guaranteeing fast initialization.

The paper is structured as follows. The next section discusses related work. After that, Sec. 3 presents an overview of our tracking framework. Sec. 4 then introduces our proposed ICP tracking approach, and Sec. 5 explains how it is integrated into the entire tracking system. Finally, Sec. 6 presents experimental results.

2. Related Work

Computer vision approaches are gaining in importance for mobile robot applications due to their capability to extract semantic information about the surrounding scene. Recently introduced vision based object detection approaches [11, 7] enabled the development of robust tracking-by-detection frameworks [1, 17, 9, 25, 15, 3] with various strategies for solving the data association problem [17, 26, 27, 15].

For real-time applications, there is a strong interest in reducing the computation time of the object detector, especially for automotive scenarios. A common strategy is to apply the detector only to those image regions which are likely to include the target object. Different approaches for extracting the ROIs were proposed based on motion [8], texture content [23], or (as already mentioned) stereo depth [12, 3]. In contrast to [12, 3, 4] we do not evaluate each ROI by the detector in every frame, but propagate the ROIs over time and apply the detector only to newly appearing ROIs.

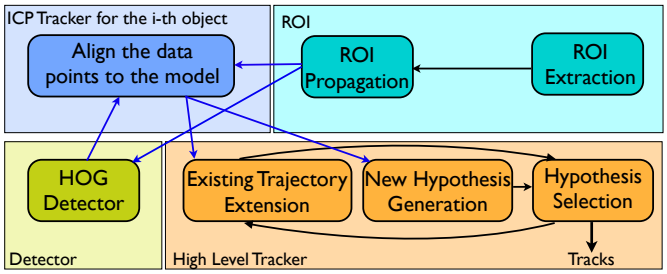


Figure 2: Overview of the different interactions between the components of our tracking system. Blue arrows indicate the interaction between the individual components. Black arrows are used representing the interaction within the components.

The idea to use depth information for tracking pedestrians has been applied in several approaches before. Arras *et al*. [2] present a pedestrian tracking approach based on single-plane scanner data by detecting and tracking legs separately with Kalman filters, forming a multi-hypothesis set. The high-level tracks which consist of two legs are extracted from the multi-hypothesis set solving the problem of occlusion and self-occlusion. However, in our approach we have to deal with dense stereo data, which comes with much higher measurement uncertainties than laser data. Most directly related to our approach, Feldman *et al*. [10] also propose an ICP based method for multi-object tracking in sports scenarios based on laser range data. This approach requires 4 calibrated laser scanners around the sports field to allow the extraction of the complete shape for each of the players. Additionally, a fixed elliptic shape model is assumed for detection and ICP tracking. However, this approach is not suitable for a mobile scenario, where only partial depth information from a single, front-facing stereo camera rig is available.

3. System Overview

Fig. 2 presents an overview of our complete tracking framework. The system is divided into four major components, whose interaction we will describe in the following: ROI candidate generation and propagation, object detection, ICP tracker, and a high level tracker.

The overarching goal of our framework is to avoid redundant detector evaluations in a spatial, as well as in a temporal context. By focusing on ROIs, the detector is applied only to relevant parts of the image, thus reducing the computation time significantly. In addition, the propagation of ROIs in time allows us to limit detector evaluations only

to newly appearing ROIs. The cheap ICP tracker, which solves short-term data association for already detected persons, takes over the role of the detector in supplying precise object location measurements to the high-level tracker. Through this combination, we reduce the number of detector evaluations to a minimum, such that the whole tracking system runs at more than 10 fps, while reaching state-of-the-art tracking accuracy.

Briefly stated, the system components interact as follows. In each frame, we first compute the ROIs by projecting the stereo 3D points onto the estimated ground plane and extracting the modes of the resulting distribution. Next, each extracted ROI is associated with the ROIs from the previous frame, which are propagated using constant-velocity Kalman filters. The ROIs that could not be associated are assumed to have newly appeared in the scene and are evaluated by the detector. Successful detections are passed to the ICP tracker, which computes a 3D model for each detection. This model is represented by the 3D points that are within a pedestrian-sized cylinder placed on the foot point of the detection on the ground plane (see Fig. 3(b)). For already existing trajectories, the ICP tracker extracts the 3D points that are located within a cylinder at the modes of the 2D grid map within the trajectory's Kalman filter prediction covariance. It then aligns the model points from the previous frame to the newly extracted data points using ICP (see Fig. 3(c,d)), resulting in a precise estimate of the new object location. From this new location, a new virtual detection is generated by projecting the position back to the image and transmitting it to the high level tracker. The detections passed to the high level tracker are employed for generating new tracks and extending the existing tracks based on an Extended Kalman Filter (EKF) using a pedestrian specific motion model.

As only few small ROIs that newly arise in the scene need to be evaluated by the detector and the main tracking work is handled by our fast ICP tracker, the entire system is very fast. The whole tracking system runs at more than 10 fps, while reaching state-of-the-art tracking accuracy. Our approach is based on stereo depth information, which is in many cases already available through dedicated sensors (*e.g.*, Microsoft's Kinect) or hardware processing solutions (*e.g.*, [22]). For our experiments, we used the depth estimation approach by Geiger *et al.* [13] that runs at 10 fps on a single CPU. In addition, we use visual odometry for estimating the camera position, and we compute the scene ground plane in each frame. For both tasks, there are also real-time approaches available [18, 16]. In order to evaluate our framework in a comparable setting, we however use the odometry and ground plane estimates provided with the datasets of [9].

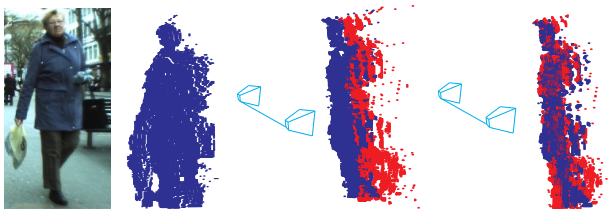


Figure 3: (a) Detection from the color image. (b) Front view of the data points in 3D. (c) The model points (red) and the data points (blue) before the ICP alignment. (d) The result of the ICP algorithm showing a correct alignment of the data and model points.

4. ICP Tracking

The ICP (Iterative Closest Point) algorithm [5, 6] is a popular method for aligning two three dimensional models based on their geometry. The goal is, given two points clouds $\mathcal{M}_k = \{\vec{m}_i\}_{i=1}^{\mathcal{N}_\mathcal{M}}$ and $\mathcal{P} = \{\vec{p}_i\}_{i=1}^{\mathcal{N}_\mathcal{P}}$ to iteratively revise the rotation and the translation which minimizes the alignment error between the point clouds:

$$\mathcal{E} = \sum_i \|\vec{p}_i^{NN} - \boldsymbol{R}\vec{m}_i - \boldsymbol{t}\|^2 \tag{1}$$

where the point $\vec{p}_i^{NN} \in \mathcal{P}$ is the closest point to $\vec{m}_i \in \mathcal{M}_k$ according to some distance function d:

$$\vec{p}_i^{NN} = \underset{\vec{p}\in\mathcal{P}}{\operatorname{argmin}}\{d(\vec{p}, \vec{m}_i)\} \tag{2}$$

In each iteration, first the closest points are computed and then the rotation and translation that minimize Eq. 1 are applied to the points $\mathcal{M}_k$ resulting in a new point cloud $\mathcal{M}_{k+1}$. The iteration is repeated for the new point cloud $\mathcal{M}_{k+1}$. Finally, the rotation and the translation of each iteration step are accumulated resulting in a final transformation.

Considering our goal to track pedestrians we can ignore the rotation estimation as pedestrians are assumed to move upright and in addition in our case the depth information is only available from one viewpoint. This constrains the ICP algorithm to approximate only the translation between two point clouds as follows:

$$\mathcal{E} = \sum_i \|\vec{p}_i^{NN} - \vec{m}_i - \boldsymbol{t}\|^2 \tag{3}$$

In each iteration step the mean distance of all corresponding (closest) points is used in order to update $\mathcal{M}_k$. We found that already after 3 iterations the points $\mathcal{M}_k$ are well aligned with the points $\mathcal{P}$ as illustrated in the example in Fig. 3.

The ICP tracker consists of two steps which are iteratively repeated.

1. For each initial detection we generate a 3D model. The 3D model is represented by the 3D points which are

sampled from a cylinder placed on the 3D position of the detection (see Fig. 3b). The cylinder size roughly approximates a person size with a radius of $0.35m$ and a height of $2.0m$.

2. The model points are employed for computing the new position of the person in the next frame by aligning them to the data points, applying the presented ICP algorithm. The new position is then backprojected to the image generating a new detection bounding box which is fed back to the high level tracker. This new detection bounding box is treated in the next frame again as the initial detection bounding box and we continue with the step 1.

As the ICP tracker cannot recognize occlusions or a person leaving the scene from the raw depth data, it can easily get stuck on some background area. However, this divergence of the ICP tracker can be detected by the high level tracker which associates the output, the 2D bounding box from the ICP trackers to the trajectories. Due to occlusion the computed detection from the ICP tracker will surely violate the motion or appearance model of the current trajectories and will not be associated to any of them. Thus in each frame the ICP trackers whose detection could not be assigned to any of the existing trajectories are terminated by the high level tracker.

5. System Realization

This section describes the realization of our framework's individual components in more detail.

5.1. Depth based ROI Generation

The stereo range data provides an important prior information for the location of vertical objects in the scene. This allows us to constrain the execution of the computationally expensive detector only to small regions and few scales of the image.

For extracting the ROIs, the 3D points of the scene are projected onto the ground plane to form a 2D histogram. The bins of the histogram are weighted by the distance to the camera and are thresholded in order to avoid noisy regions. The weighting is necessary since the objects that are further away consist of fewer points and would therefore be rejected in the further processing. The final ROIs in 3D are the connected components on the grid map as seen in Fig. 4. The position of each ROI is specified by the center of mass of the corresponding points. Additionally, we keep the width of the ROI and also the histogram modes. The modes are required for the low level tracker in order to distinguish between two or more pedestrians walking close together which will produce a single connected ROI.

For each ROI in 3D we set a rectangle at the center of mass of the ROI with the width of the ROI and a height of 2 meters positioned parallel to the camera. This rectangle is projected to the image in order to obtain the corresponding image region that is evaluated by the detector.

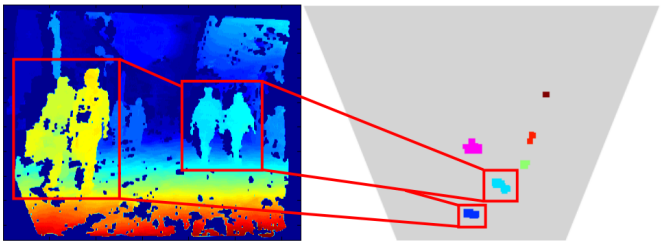


Figure 4: Extraction of the ROIs (right) by projecting the 3D points from the depth map (left) to a ground plane.

5.2. Object Detection

As pedestrian detector we employ our GPU based HOG (Histograms of Oriented Gradients) detector [7]. With our implementation [24] we can achieve the same detection performance as reported in the original paper [7] and it requires only 40ms for a 640 $\times$ 480 image.

In contrast to pure tracking-by-detection approaches [9] we run the detector only on few small ROIs and only for some scales. If a new ROI shows up in a frame, the detector is run on this ROI evaluating only 5 scales instead of 27 as in the original approach. We already know the rough scale by dividing the height of the back-projected ROI from 3D to the image by the detector window height. As the height of the pedestrian in the ROI is not known we consider additionally two scales above and two scales below the computed scale in order to assure the detection of the pedestrian.

The evaluation of a ROI through the detector requires on average only 2-3ms. In addition our ROI system is able to also detect pedestrians that are smaller than 128 pixels (64x128 pixel detection window constrains the smallest possible detection). In particular for pedestrians which are far away from the camera the backprojected height h of the bounding box in the image will be smaller than 128 pixels resulting in a scale of $s = \frac{128}{h}$. The scale s is larger than 1, thus the ROI will be upscaled and the corresponding person can be found by sliding over the upscaled ROI with the standard HOG window. For achieving equivalent performance with the sliding window detector, the image needs to be upscaled by factor of two to the size of 1280 $\times$ 960. Processing such an image with our GPU detector would require more than 180ms making it not applicable for real-time systems.

5.3. ROI Propagation

Processing ROI association allows us to reduce the detector time to a minimum. In particular in each frame, the newly extracted regions of interest need to be associated with the ROIs from the last frame. To this end we apply a Kalman filter with a constant-velocity model starting from

each ROI from the previous frame. The system uncertainty is propagated from frame to frame for each ROI and is employed for finding the new predicted position. In addition the frame number when the ROI was last evaluated by the detector is stored along with the associated region of interest. This step is relevant for a periodic reinitialization for alleviating the problem of false negatives.

For the new extracted ROIs that could not be associated, we run the detector and if a detection is found a new ICP tracker is started for this detection.

5.4. High Level Tracking Model

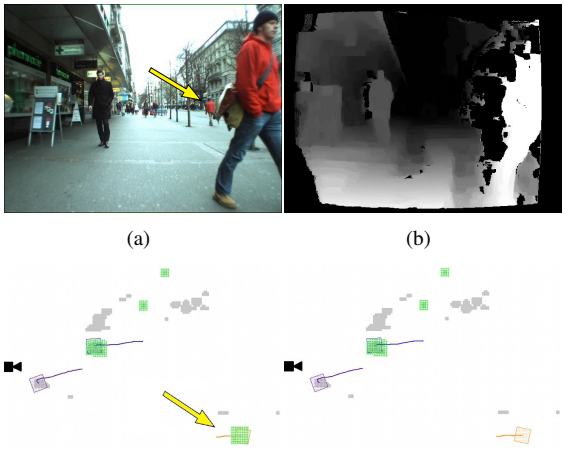
As a high level tracker we employed a simplified version of the robust multi-hypothesis tracking framework presented by [17].

In the pure tracking-by-detection approach in each frame the detections are transformed into global 3D world coordinates by projecting the foot point of the bounding box to the ground plane using the scene geometry information. These 3D positions are accumulated over time to multiple competing hypotheses set. The trajectory hypotheses are pruned to a final set that best represents the scene using a model selection framework.

Trajectory Hypothesis Generation. For linking the detections on the ground plane we employ an Extended Kalman Filter (EKF) with a constant-velocity model. In each frame when new detections are available we run two trajectory generation processes. Firstly, we try to extend the existing trajectory by the new evidence. Secondly, for each new evidence we generate new trajectory hypotheses using the Kalman filter backwards in time up to 100 frames. This allows us to bridge occlusions. Due to the fact that the new observations are utilized for both processes extension of the existing trajectories and generation of new trajectories, each observation can be potentially assigned to two competing hypotheses.

Pruning of Hypothesis Set. The trajectory hypothesis generation process outputs an over-complete set of trajectories. Each hypothesis is ranked based on the likelihood of the assigned evidence under motion (pedestrian specific constant velocity motion model) and appearance model (represented as an RGB color histogram). Trajectory hypotheses compete for the evidence through penalties if they overlap. The final consistent set is obtained using model selection in a Minimum Description Length framework, as shown in [17].

Person Identities. Since the model selection procedure may choose a different hypothesis for a trajectory, either the extended or the newly created one, we need to assure that a consistent person ID is assigned to a selected trajectory. Hence we propagate the identities by computing the overlap between the trajectories found at the current frame with the trajectories from the previous frame.



(a) (b)

(c) Using occlusion detection (d) Without occlusion detection

Figure 5: Example showing the shadowing problem in stereo range data and our strategy to resolve it using explicit occlusion detection. (a,b): No regular ROI is generated for the person marked with a yellow arrow due to shadowing artifacts in the depth map. (c,d): The green areas are the ROIs which were verified by the detector in the current frame. Without occlusion detection, the person is missed. In contrast, the occlusion detection module prompts the detector to directly verify the area where the person reappears, although there is still no depth information available.

5.5. Occlusion Handling

When dealing with stereo range data (as in our case), we have to deal with the well-known shadowing problem, which occurs for the part of the images which are visible only from one camera. This problem is visualized in Fig. 5(b). Due to the lack of depth information in the area behind the closest person to the camera, it is not possible to find a region of interest for the person marked with the yellow arrow. In example in Fig. 5(b), it takes three further frames until the shadow disappears and the marked person is visible in the stereo range data. In order to be able to associate already existing tracks after an occlusion as fast as possible and avoid losing the trajectory (after 15 frames without observation, the trajectory is removed), we propose to predict possible person-person occlusions and create a ROI when those are over. For this, we propose to project the 3D prediction of the EKF of each tracked person into the image and compute the bounding box overlap using the intersection-over-union criterion. If the overlap is above 0.5, then the occlusion is likely to occur and we mark the person as occluded. In each further frame, we check for all occluded persons if they become visible again within 15 frames. If a person reappears, we create a ROI at this position and run the detector to evaluate this region and to revise the tracker, giving a new observation for the EKF.

In Fig. 5(c),(d) we show the ROIs that are evaluated if the person marked with the yellow arrow reappears after occlusion. As can be seen in Fig. 5(c), the result of occlusion

detection directs the detector to evaluate the area where the person reappears, although there is no ROI extracted from depth. In contrast, Fig. 5(d) shows the result without using the described method, where the person remains undetected.

5.6. Reinitialization

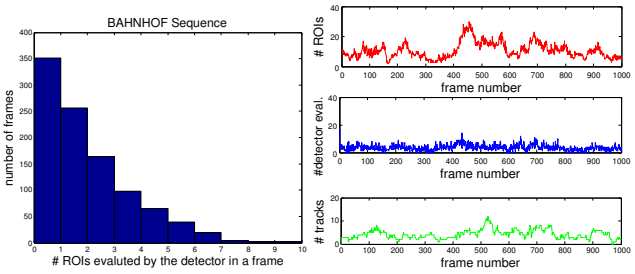
Due to false negatives of the detector some of the pedestrians could be missed and not tracked over the whole time if newly appeared ROIs were evaluated only once. For robustness, we propose to periodically re-trigger the object detector for ROIs which did not have a positive detector response once they appeared in the scene. In particular during the ROI association process we also propagate the frame when a ROI emerges into the scene and was consequently evaluated by the detector. For ROIs with negative initial detector response we execute the detector once again after five and ten frames. If the detector outputs a detection we continue with the ICP tracker, if the detector response is negative we continue propagating the ROI, but do not run any further detector evaluations on it.

5.7. Consistency Checks

For robustness reasons it is necessary to check the consistency of the tracking results of the ICP tracker. The consistency check is performed by the high-level tracker by associating the resulting detections from the ICP trackers. If a detection from the ICP tracker is physically inconsistent in motion and does not fit the appearance with at least one of the existing trajectories, the high level tracker terminates the particular ICP tracker. For each trajectory which could not be extended due to divergence of the ICP tracker, the high-level tracker generates a ROI at the current position of the trajectory. The generated ROI is treated as a newly emerged ROI, which is consequently evaluated by the detector and attempted to be associated with new ROIs in the next frame. This process assures that if the ICP tracker fails, the trajectory can still be robustly extended. Obviously, only those tracks are considered here that are not labeled as occluded.

6. Experimental Results

We experimentally evaluate our approach on three challenging video sequences provided by [9]. All sequences were captured using a child stroller carrying a stereo rig. The image data was acquired at 13-14 fps and a resolution of 640 × 480. The BAHNHOF sequence was captured on a crowded side walk and contains 999 frames with 5193 annotations. The JELMOLI sequence was captured in a busy pedestrian zone and contains 999 frames with 446 frames that are annotated. The SUNNY DAY sequence was acquired on a sunny day on a crowded side walk and contains 999 frames out of which 354 are annotated. For all sequences the structure-from-motion localization and ground



(a) (b)

Figure 7: (a) Histogram of the number of ROI evaluations by the detector per frame for the BAHNHOF sequence. (b) number of extracted ROIs in each frame (red), number of ROIs that were evaluated by the detector (blue), and number of final tracks estimated for each frame (green).

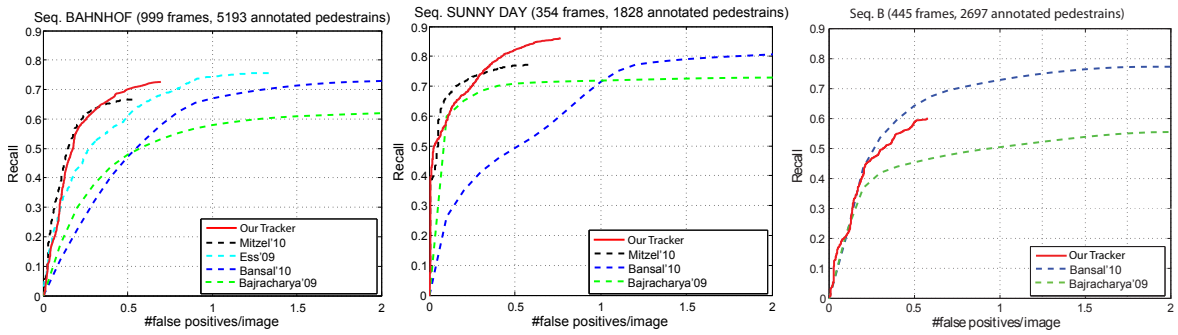
baseline	speed in fps	recall@0.5 fppi
pure t-by-d 640 × 480	14.59	0.60
pure t-by-d 1280 × 960	4.62	0.69
no ICP, eval. all ROIs	6.54	**0.70**
ICP, number of sampled points	**speed in fps**	**recall@0.5 fppi**
ICP, 20 points	11.62	0.67
ICP, 50 points	11.36	0.69
ICP, 100 points	10.86	**0.70**
ICP, 500 points	7.23	0.69
ICP, all points	1.09	**0.70**

Table 1: Comparison of our proposed ICP tracker results to several baselines on the BAHNHOF sequence. We report the effect of the number of (randomly sampled) ICP points on the final tracker run-time and accuracy.

plane estimates are made available by [9]. For the depth estimation we used the fast and robust algorithm presented by [13] (10 fps on a single CPU) .

Quantitative Performance. For assessing the performance of our tracking system, we applied the evaluation criteria from [9]. To this end the tracked bounding boxes are compared to manually annotated ground truth bounding boxes in each frame. A bounding box is assumed to be correct if the intersection-over-union overlap with a ground truth bounding box is greater than 0.5. Fig. 6 presents the performance curves in terms of recall vs. false positives per image (fppi) for three sequences. As can be seen, our approach achieves state-of-the-art performance. For comparison, we also provide the curves reported by [9] (only BAHNHOF), [19](only BAHNHOF and SUNNY DAY) and [4, 3]. Note that [4] did not use the original annotation files provided by [9] but created their own, leaving out some hard test cases.

Computational Performance. Most of the computation time in standard tracking-by-detection approaches is required for the object detector, especially if more than one object class/view needs to be detected. In order to evaluate the effect of our proposed ICP tracker on reducing this



(a) (b) (c)

Figure 6: Quantitative tracking performance of our approach compared to different baselines on the BAHNHOF, SUNNY DAY and JELMOLI sequences from [9]. The results show that our approach can reach state-of-the-art performance. For all three test the same parameter sets were used.

Figure 8: Example results on the test sequences SUNNY DAY (top) and BAHNHOF (bottom).

computation, we performed the following timing experiments (using a machine with Intel Core2 Quad CPU Q9550 @ 2.83GHz processor, 8GB RAM, and an Nvidia GeForce GTX 280 graphics card).

Running the ICP tracker with all available model and data points is very time consuming and not really necessary, as shown in Tab. 1. Here, we illustrate the effect of using only a fixed number of randomly sampled points from the model and data for the ICP tracker. With only 100 points, we can already reach state-of-the-art tracking accuracy (recall of 0.7 @ 0.5 ffpi on the BAHNHOF sequence) with a speed of 10.86 fps.

As a baseline, we compare the run-time of our approach to a pure tracking-by-detection system. To this end, we run our GPU-accelerated object detector (requires 40ms for a 640×480 image) over the entire image in each frame and use the high-level tracker for data association. The frame rate is significantly higher (14.59 fps) compared to the proposed hybrid tracker (10.86 fps), but the recall decreases considerably (from 0.70 to 0.60 @ 0.5 fppi), since only pedestrians larger than 128 pixels (the height of the sliding window of the HOG detector) could be detected. When upscaling the image to twice its original resolution, the missing pedestrians can be detected (recall of 0.70 @ 0.5 fppi); however, the frame rate drops considerably to 4.6 fps.

For comparison, we also report an experiment where we evaluate the effect of the ROI propagation scheme. Instead of running the ICP tracker starting from an initial detection, the detector is run in each frame for all ROIs. As expected, we reach high recall in this test (0.70 @ 0.5 fppi), but the frame rate drops significantly to 6.54 fps, since we redundantly run the detector for each ROI in each frame instead of propagating the information whether the ROI contains a person or not. In Fig.7 we illustrate the triggering rate of the detector for the BAHNHOF sequence. Note that in $2/3$ of the frames, the detector runs only at most for two small ROIs. On average the detector is executed for 2.38 ROIs per frame.

Fig.7(b,c) presents the relation between the number of detector evaluations, number of all ROIs and number of

valid tracks. As expected, in the part of the scene with many tracks also the number of detector evaluations increases. Due to occlusion and clutter in crowded parts of the scene, the ICP tracker diverges, causing the consistency check to fail. Thus, the high-level tracker generates additional ROIs that are evaluated in order to achieve robust tracking performance and not to lose tracks.

Qualitative Evaluation. Finally, Fig.8 shows some qualitative results achieved on the BAHNHOF and SUNNY DAY sequences. It can be seen that our system is able to track most of the visible persons correctly keeping correct person identies. In addition to the tracking bounding box, we visualize the depth information that was used for computing the ROIs (in green inside the box). The ROI foot points are plotted as red points on the ground plane. Detailed results can also be found in the supplementary material.

7. Conclusion

We have presented a hybrid framework for mobile multi-person tracking. Our approach reduces the use of a computationally expensive detector by a combination of ROI propagation, low-level ICP tracking and a high-level tracker. As our experimental evaluations show, we can reach state of the art tracking performance with this combination at a runtime that is suitable for real-time applications.

In the future we plan to apply the ICP approach to more complex tracking scenarios including also other object classes such as cars, where several different viewpoint detectors need to be evaluated for each ROI. For such cases, we expect the run-time benefit to be even larger. In addition we plan to combine our approach with ROI selection techniques [20].

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).

References

[1] M. Andriluka, S. Roth, and B. Schiele. People Tracking-by-Detection and People Detection-by-Tracking. In *CVPR*, 2008.

[2] K. Arras, S. Grzonka, M. Luber, and W. Burgard. Efficient People Tracking in Laser Range Data Using a Multi-Hypothesis Leg-Tracker with Adaptive Occlusion Probabilities. In *ICRA*, 2008.

[3] M. Bajracharya, B. Moghaddam, A. Howard, S. Brennan, and L. Matthies. A Fast Stereo-based System for Detecting and Tracking Pedestrians from a Moving Vehicle. *IJRS*, 2009.

[4] M. Bansal, S. H. Jung, B. Matei, J. Eledath, and H. S. Sawhney. A real-time pedestrian detection system based on structure and appearance classification. In *ICRA*, 2010.

[5] P. J. Besl and H. D. Mckay. A method for registration of 3-D shapes. *PAMI*, 1992.

[6] Y. Chen and G. Medioni. Object modeling by registration of multiple range images. In *ICRA*, 1991.

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

[8] M. Enzweiler, P. Kanter, and D. Gavrila. Monocular Pedestrian Recognition Using Motion Parallax. In *Intel. Vehicles Symp.*, 2008.

[9] A. Ess, B. Leibe, K. Schindler, and L. Van Gool. Robust Multi-Person Tracking from a Mobile Platform. *PAMI*, 2009.

[10] A. Feldman, M. Hybinette, T. Balch, and R. Cavallaro. The Multi-ICP Tracker: An Online Algorithm for Tracking Multiple Interacting Targets. submitted, www.cc.gatech.edu/-tucker/Papers. 2011.

[11] P. Felzenszwalb, B. Girshick, D. McAllester, and D. Ramanan. Object Detection with Discriminatively Trained Part-Based Models. *PAMI*, 2010.

[12] D. Gavrila and S. Munder. Multi-Cue Pedestrian Detection and Tracking from a Moving Vehicle. *IJCV*, 2007.

[13] A. Geiger, M. Roser, and R. Urtasun. Efficient Large-Scale Stereo Matching. In *ACCV*, 2010.

[14] D. Geronimo, A. Sappa, D. Ponsa, and A. Lopez. 2D-3D-based On-Board Pedestrian Detection System. *CVIU*, 2010.

[15] C. Huang, B. Wu, and R. Nevatia. Robust Object Tracking by Hierarchical Association of Detection Responses. In *ECCV*, 2008.

[16] R. Labayrade and D. Aubert. A Single Framework for Vehicle Roll, Pitch, Yaw Estimation and Obstacles Detection by Stereovision. In *Intel. Vehicles Symp.*, 2003.

[17] B. Leibe, K. Schindler, and L. Van Gool. Coupled Object Detection and Tracking from Static Cameras and Moving Vehicles. *PAMI*, 2008.

[18] C. Mei, G. Sibley, M. Cummins, P. Newman, and I. Reid. A constant time efficient stereo slam system. In *BMVC*, 2009.

[19] D. Mitzel, E. Horbert, A. Ess, and B. Leibe. Multi-person tracking with sparse detection and continuous segmentation. In *ECCV*, 2010.

[20] D. Mitzel, P. Sudowe, and B. Leibe. Real-Time Multi-Person Tracking with Time-Constrained Detection. In *BMVC*, 2011.

[21] K. Okuma, A. Taleghani, N. de Freitas, J. Little, and D. Lowe. A Boosted Particle Filter: Multitarget Detection and Tracking. In *ECCV*, 2004.

[22] C. Rabe, T. Müller, A. Wedel, and U. Franke. Dense, robust, and accurate motion field estimation from stereo image sequences in real-time. In *ECCV*, 2010.

[23] A. Shashua, Y. Gdalyahu, and G. Hayon. Pedestrian Detection for Driving Assistance Systems: Single-Frame Classification and System Level Performance. In *Intel. Vehicles Symp.*, 2004.

[24] P. Sudowe and B. Leibe. Efficient Use of Geometric Constraints for Sliding-Window Object Detection in Video, www.mmp.rwth-aachen.de/projects/groundhog. In *ICVS*, 2011.

[25] B. Wu and R. Nevatia. Detection and Tracking of Multiple, Partially Occluded Humans by Bayesian Combination of Edgelet Part Detectors. *IJCV*, 2007.

[26] L. Zhang and R. Nevatia. Global Data Association for Multi-Object Tracking Using Network Flows. In *ECCV*, 2008.

[27] T. Zhao, R. Nevatia, and B. Wu. Segmentation and Tracking of Multiple Humans in Crowded Environments. *PAMI*, 2008.