Taking Mobile Multi-object Tracking to the Next Level: People, Unknown Objects, and Carried Items

Dennis Mitzel and Bastian Leibe

Computer Vision Group, RWTH Aachen University, Germany

Abstract. In this paper, we aim to take mobile multi-object tracking to the next level. Current approaches work in a *tracking-by-detection* framework, which limits them to object categories for which pre-trained detector models are available. In contrast, we propose a novel *tracking-before-detection* approach that can track both known and unknown object categories in very challenging street scenes. Our approach relies on noisy stereo depth data in order to segment and track objects in 3D. At its core is a novel, compact 3D representation that allows us to robustly track a large variety of objects, while building up models of their 3D shape online. In addition to improving tracking performance, this representation allows us to detect anomalous shapes, such as carried items on a person's body. We evaluate our approach on several challenging video sequences of busy pedestrian zones and show that it outperforms state-of-the-art approaches.

1 Introduction

The capability to reliably track people in street scenes from a mobile platform is important for many applications in mobile robotics and intelligent vehicles. In recent years, a number of *tracking-by-detection* approaches have been proposed for this goal [1–7], achieving remarkable tracking performance. However, those approaches are naturally restricted to tracking objects for which pre-trained detector models are available.

In order to understand the behavior of people, it is also important to recognize and track other objects in their surroundings. In practical scenarios, this includes a large variety of objects such as bicycles, child strollers, shopping carts, trolleys, or wheelchairs (see Fig. 1 for some examples). In addition, people's motion is often affected by accessories such as backpacks, shopping bags, walking aids, or other personal items. While some of those object categories are sufficiently frequent and distinctive that the effort pays off to learn generic detectors for them in an offline fashion [8], this is typically not the case in general. There is an almost endless variety of baggage items that one would need to learn models for, and this variety may change from one place to the next and according to quick-lived fashions. We therefore need to develop methods that can detect and track also novel object types and learn models for them online.

This problem is not trivial to solve, because it necessitates a solution for a more fundamental issue, namely the question to decide what is an object? When interpreting text, it is easy to detect that there is an unknown word, but in order to do the equivalent in vision, one needs to segment out the object first, *i.e.*, to determine its precise extent in space and time from the input video stream. This requires either familiarity with the unknown object's appearance (which we explicitly assume not to be known a-priori) or

A. Fitzgibbon et al. (Eds.): ECCV 2012, Part V, LNCS 7576, pp. 566–579, 2012.

(a) (b) (c) (d)

Fig. 1. In addition to pedestrians, real-world scenarios contain a large variety of other dynamic objects (a), for which there are usually no generic detectors available. We propose an approach that can reliably track those objects, together with the accompanying pedestrians, from a mobile platform (b). Our approach builds up 3D shape models for all tracked objects (c). By comparing those models to a statistical model of typical pedestrian shapes, it additionally allows us to create hypotheses about the presence of carried items (d, shown in red).

an interpretation of enough of the object's surroundings to determine that there is an image region that still needs to be explained.

In this paper, we propose a novel 3D tracking approach that addresses those issues together. In contrast to most existing approaches for mobile multi-object tracking in street scenes [1–7, 9], we adopt a rigorous *tracking-before-detection* strategy. We make use of noisy stereo depth estimates to extract regions-of-interest (ROIs) in the input images and robustly segment them into candidate objects. Each such region is then tracked independently in 3D using a model-based iterative-closest-point (ICP) [10] tracker. For each tracked object, we then pass the corresponding image ROI to an object detector [8] for classification into pedestrians and other objects. As a result, we can not only track a large number of pedestrians and unknown objects, but we additionally save computation, since the person classifier needs to be applied only once for an entire track.

At the core of our approach is a novel, compact 3D representation that allows us to integrate and refine both volumetric and surface information about tracked objects over time. This representation makes it possible to robustly track a large variety of objects despite high levels of noise in the stereo input. As a side result, our approach builds up an integrated shape model of the tracked objects that can be used for further analysis. In a final step, we use the online-learned models in order to analyze the shape of tracked persons in more detail. By comparing their observed shapes with a learned statistical shape template of unencumbered pedestrians, our approach can generate plausible hypotheses about the presence of carried items on their bodies. To our knowledge, ours is the first approach to render such detailed shape analysis possible for a vision system on-board a mobile platform.

Related Work. The first challenge in any tracking system is to reliably segment objects of interest from their background. For mobile pedestrian tracking, this is usually done using the *tracking-by-detection* paradigm [1, 2, 5–7, 11], where the initial segmentation task is performed by an object detector and tracks are then extended by additional detections. In some approaches, the detector is only used for initializing new tracks, which are then propagated by low-level trackers [3, 4, 9]. Common to all of those approaches is that only objects are tracked which have first been verified by the detector.

The contrary paradigm of *tracking-before-detection* was first used for tracking tasks in radar data [12, 13], where measurements were probabilistically integrated over time before passing them to a classifier in order to compensate for low signal-to-noise ratios. Similarly, tracking-before-detection has been used for driver assistance systems based on stereo input (*e.g.*, [14]). The difficulty of obtaining robust and accurate stereo measurements has however often restricted such approaches to structured driving environments, where objects of interest are usually well separated from each other and their backgrounds, whereas detection-based approaches are usually used for pedestrian tracking [15]. The development of highly accurate dense 3D laser range sensors (LIDAR) has made it possible to use tracking-before-detection approaches also for more challenging autonomous driving scenarios [16, 17]. However, the excessive cost of current LIDAR sensors (list prices 30k-75k US$) restricts more wide-spread use. In contrast, the approach proposed in this paper is able to operate in very challenging pedestrian zone scenarios, while being robust enough to work on cheap stereo vision input.

Stereo depth-based region-of-interest (ROI) extraction is often used in order to speed up pedestrian detection and reduce the number of false positive detections in mobile scenarios [3, 4, 15, 18]. This is usually based on the assumption that objects of interest occur on a ground surface, and various road shape models have been proposed to remove the ground points [15, 19]. Our ROI extraction approach is similar to the ones from [3, 4], but it is especially optimized for robust region extraction in busy scenarios.

After extracting ROIs for candidate objects, we continue tracking them in 3D using ICP [10]. Several approaches have been proposed in the past for ICP-based multi-person tracking [20, 21]. Our approach differs from them in that it builds up a compact 3D model for each tracked object capturing both surface and volumetric properties. This allows our approach to integrate and refine the noisy stereo measurements over time, resulting in more robust tracking performance. Recently, methods have also been proposed for learning object categories from tracked LIDAR point clouds [22, 23]. Our proposed 3D representation makes such algorithms applicable to noisy stereo data, while simplifying the learning task by providing valuable integrated shape information in addition to appearance and trajectory data. The approach of [24] represents unknown object shapes (extracted with the help of fixation cues of a robotic stereo vision system) by 3D Gaussians. Our 3D representation preserves more detail about surface shape in addition to observed shape variances, which helps us cope with articulated objects.

For detecting carried objects in static surveillance camera footage, [25] have proposed an approach that learns view-specific temporal templates of an unencumbered pedestrian's occupied area and compares this template to the segmented area of tracked objects obtained by background modeling. We take inspiration from their approach, but generalize it to the case of a moving camera, where segmentation and model construction is a much harder problem.

The rest of the paper is structured as follows. Sec. 2 gives an overview of our processing pipeline. Sec. 3 then introduces our compact 3D object representation, and Sec. 4 describes how it is used for tracking-before-detection. Sec. 5 explains our approach for carried item detection. Finally, Sec. 6 presents experimental results.

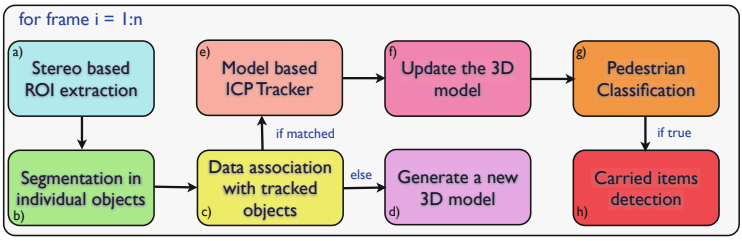


Fig. 2. Overview of the different steps of our approach

2 Overview

Fig. 2 shows a system-level overview of our proposed 3D tracking framework. Given a new stereo image pair and its corresponding depth map, we first generate ROIs (a) by projecting the 3D points within a certain height corridor onto the ground plane, excluding wall regions, and segmenting the resulting projection image into distinct object regions (b). We then perform data association (c) with a nearest-neighbor filter in order to associate new candidate ROIs with existing objects based on ROI bounding box overlap in the ground projection. For each unmatched object, we build up a compact 3D model (d) from the 3D points within its ROI and start a new track. In the following frames, we then use this 3D model in order to precisely register the tracked object surface to the ROI's 3D point cloud using model-based ICP (e). After each successful alignment, we update the model with the new 3D points, building up an uncertainty estimate for each surface point (f). When the tracked object comes into detection range, we apply a pedestrian detector to its ROI in order to classify the tracked object as person or non-person (g). Note that this has to be done only once for each track. Finally, we perform detailed shape analysis for tracked persons close to the camera (h). By comparing their accumulated 3D models with a learned model of unencumbered pedestrian shapes, we create hypotheses for carried items (*e.g.*, backpacks or shopping bags) on their bodies.

Dealing with Camera Motion. In order to apply the above procedure robustly for a mobile setup, we additionally perform the following steps, which are, however, not a contribution of this paper. We track the camera position and orientation over time using stereo visual odometry [26]. This allows us to perform object tracking in 3D world coordinates, which considerably improves the robustness of the data association and registration steps. Since standard visual odometry quickly degenerates in crowded scenarios, we create a feedback loop with the tracker in order to exclude regions on moving objects from visual odometry computation, as proposed by [2]. This step significantly improves the robustness for crowded scenarios. In addition, we feed the camera pose estimates from each frame into a Kalman filter, which allows us to bridge frames where no reliable features could be found after the exclusion step.

3 3D Object Representation

Our goal is to capture the approximate 3D shape of arbitrary objects in order to facilitate accurate 3D tracking and more detailed shape analysis. A main difficulty when trying

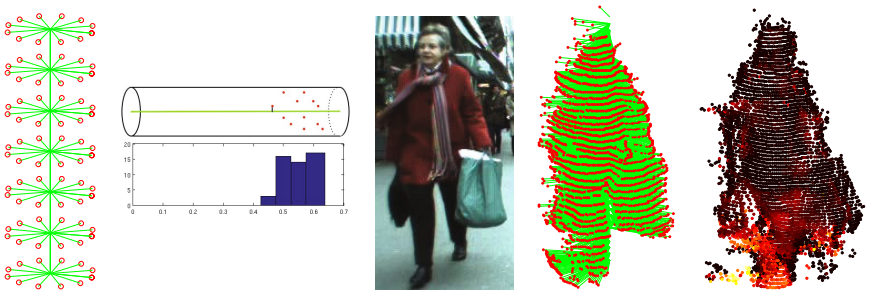


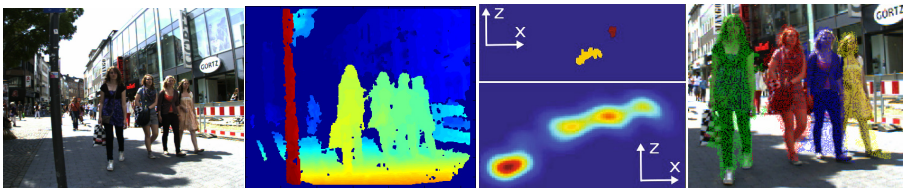
Fig. 3. We represent arbitrary 3D objects by a novel *Generalized Christmas Tree* (GCT) model, a cylindrical shape model composed of a vertical center axis and several layers of equally spaced horizontal rays (a). Along each ray, we accumulate observed 3D points in a distance histogram (b). This representation naturally adapts to the shape of tracked objects (c), allowing us to use median points (d) and variances (e) of each ray for robust model-based tracking. (For better visualization, only a subset of the height layers are shown).

to do this from stereo depth input is that the raw depth estimates are very noisy and depth resolution diminishes with distance. (For example, in our data, a pedestrian at 9m distance may cover only 1-2 disparity levels). To work with such data, we therefore need to integrate shape information over time to smooth out noise. Since our aim is to develop an approach that scales well to a large number of objects, on the other hand, it is important that the representation is compact, capturing the essence of a novel object shape in a form that is memory efficient and easy to update.

In this paper, we propose a novel, compact shape representation that fulfills those goals, called a *Generalized Christmas Tree* (GCT). As visualized in Fig. 3, this model consists of a vertical center axis and several height layers of equally sampled horizontal rays emanating from this axis. Each ray stores the distance distribution of observed 3D surface points within a certain cylindrical cross-section. This way, the GCT captures both a robust estimate for the most likely object surface, represented by the median points for each ray, and the variance along each ray caused by noise and articulations.

To construct a GCT from an ROI, we cast radial rays from the center of the ROI over a fixed number of discrete height levels (see Fig. 3(a,d)). For each ray, we consider all 3D points that fall inside a cylindrical volume and select the closest point to the ray as their representative. This point is projected onto the ray and only the distance from the center to the projected point is stored. When the model is updated in the following frames, additional points are associated to the ray. In order to allow tracking over long sequences with a fixed memory footprint, we store the distances in a histogram Fig. 3(b)).

This is clearly an approximation, as not all object shapes can be accurately captured by a GCT. As only angular distances to the farthest surface are stored, some shape details, such as the space between a person's arms and its body, are lost in the representation. Still, the approximation is reasonable for a large variety of objects encountered in street scenes. In addition, the representation is very compact, requiring only the storage of a fixed number of histogram bins for each ray, which is far less information than the alternative of storing a full point cloud for each frame in which the object is observed.



(a) (b) (c) (d)

Fig. 4. Visualization of our stereo ROI extraction and segmentation procedure. (a) Original image; (b) depth map; (c) projection onto ground plane. In the initial ground projection (top) the four women in front are merged into a single connected component (in yellow), which our approach segments into four ROIs using Quick Shift [27]. (d) resulting point cloud segmentation.

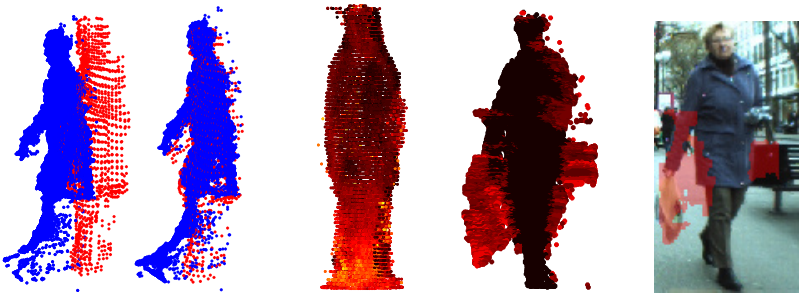
As the stereo depth image contains only information about the front surface of a tracked object, each GCT will typically only capture a partial shape. Still, by aligning the GCT with the object trajectory, we are able to grow and complete the object model whenever an object turns around. Thus, the GCT model can also capture volumetric information. When initializing a new GCT, a main uncertainty factor is the distance from the observed object surface to its center. Since the center location is not a-priori known for novel objects, we first make an initial guess, assuming an object radius of 50cm. Instead of directly building up a distance histogram for each ray, we first keep a list of the raw associated distances for several frames. When enough evidence has been accumulated, we update the representation by reconstructing a point cloud from the stored distances, re-estimating the object center to fit the point cloud dimensions, and resampling all reconstructed points into the rays to fit the shifted object center.

4 Stereo Depth-Based Tracking-Before-Detection

In the following, we present the detailed stages of our tracking pipeline.

Stereo ROI Extraction and Segmentation. As a first step, we generate a set of ROIs for potential objects in the scene from the stereo depth image (see Fig. 4). Following the approach by [4], we take a rough estimate for the location of the ground plane based on the camera height of our recording vehicle and project all 3D points within a 2m height corridor onto this plane. We collect the points in a 2D histogram and weight them according to the square distance to the camera in order to compensate for varying depth resolution. We exclude points on walls and other elevated structures by removing regions that continuously extend beyond a height of 2m. The histogram bins are thresholded for removing noise and the remaining bins are grouped into connected components using an 8-neighborhood. The resulting connected components are shown in Fig. 4(c,top).

As can be seen from the figure, people walking close to each other are still connected in the ground projection. In order to track them individually, we segment the connected components further using a smoothed version of the original histogram (Fig.4(c,bottom)). For this, we employ the Quick Shift algorithm [27], a fast variant of mean-shift. Quick Shift finds the modes of a density $P(x)$ by shifting each point x_i to the nearest neighboring point y_i with a higher density value. It is formally defined as follows:



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

Fig. 5. (a) Visualization of the ICP registration procedure for aligning model points (red) of a tracked object to new 3D points from the current frame (blue). (b) Result of the ICP alignment. (c) Visualization of the learned pedestrian shape model. Points are drawn at the median distances; the colors represent the variances. (d) Result of comparing a tracked pedestrian model to the learned shape model. The brighter the color, the larger is the difference in point distributions. (e) Result after Graph-Cut segmentation for carried-item detection (overlayed in red).

$$y_i = \underset{j:P_j > P_i}{\operatorname{argmin}} d(x_i, x_j), \;\; P_i = \frac{1}{N} \sum_{j=1}^{N} \theta(d(x_i, x_j)) \tag{1}$$

where P_i is the density estimate for point x_i with kernel function θ, and $d(x_i, x_j)$ is the distance between points x_i and x_j. In our case, we operate on a regular grid and use a fixed window (the 8-neighborhood around a pixel) as the kernel, which lets each pixel simply move to the neighbor with the highest value. We start Quick Shift for each point in the smoothed ground projection histogram and move it to its respective neighbor with maximum value until a mode is reached. The points on the way to the mode are automatically associated to this mode. As a result, we obtain a segmentation of the ROIs into individual objects, as shown in Fig.4(d).

Ground Plane Refinement. As a byproduct of the ROI estimation, which extracts all points above the ground plane, we obtain a rough segmentation of the points on the ground plane itself. We use those remaining points to obtain a more accurate ground plane estimate by fitting a plane to them using RANSAC. This refined estimate can then be used for ROI extraction in the next frame.

Data Association. In order to associate ROIs with existing tracks, we match them to each track's ROI from the previous frame. We assume ROIs to match if the intersection-over-union of their ground projection footprints is over 50%. We associate each track with the closest matching unassociated ROI and start new tracks for all ROIs that cannot be associated.

ICP Tracking and Model Update. After associating each track with a new ROI, we align the accumulated GCT object models with the ROIs' 3D points in the new frame. For this registration step, we adapt the Iterative Closest Point (ICP) algorithm [10]. Briefly stated, ICP iteratively computes the rotation $\boldsymbol{R}$ and translation $\boldsymbol{t}$ for aligning two point clouds $\mathcal{M} = \{\boldsymbol{m}_i\}_{i=1}^{|\mathcal{M}|}$ and $\mathcal{D} = \{\boldsymbol{d}_i\}_{i=1}^{|\mathcal{D}|}$ based on correspondences between closest points

$$\mathcal{E}(\boldsymbol{R},\boldsymbol{t}) = \sum_{i}^{|\mathcal{M}|}\sum_{j}^{|\mathcal{D}|} w_{i,j}\|\boldsymbol{d}_i - (\boldsymbol{R}\boldsymbol{m}_j + \boldsymbol{t})\|^2 \quad (2)$$

where $w_{i,j}$ is 1 if $\boldsymbol{d}_i$ is closest point to $\boldsymbol{m}_j$ and 0 otherwise. We restrict $\boldsymbol{R}$ to 1D rotations around the ground plane normal and use the Euclidean distance as point-to-point distance measure.

Fig. 5(a-b) shows the resulting registration procedure. For each GCT model, we first generate the 3D points at the position of the previous frame by taking the median of the distances accumulated so far, corresponding to $\boldsymbol{m}_j$ in Eq. 2 (red points in Fig. 5(a)). The blue points are the 3D points from the overlapping ROI, which correspond to $\boldsymbol{d}_i$ in Eq. 2. In order to robustly register articulated objects, we weight each GCT point by the standard deviation of its distance distribution, adapting $w_{i,j}$ accordingly. After performing the ICP registration, we obtain the translation and rotation for moving the GCT model to the new position in the current frame. Next, we update the registered GCT model with the new 3D points. For this, we again match 3D points to the model's rays, choose the closest point as representative for each matched ray, and add it to the stored distance distribution.

Object Classification. Whenever a newly generated track comes into detection range, its ROI is passed to a pedestrian detector for person/non-person classification. For this, we crop a small region around the back-projected segmented 3D points and only evaluate the detector in this region. As long as the object is then continuously tracked, no further classification is performed. In our experiments, this meant that our approach only needed to evaluate on average 0.7 small ROIs per frame with the detector (we used the DPM detector from [8]). As our results in Sec. 6 will show, this procedure still achieves competitive pedestrian tracking performance.

Safe Tracklet Termination. The proposed approach generates long tracklets, but obviously cannot track through occlusions, as it depends on the registration of point clouds. In order to cope with occlusions and temporary segmentation failures, we therefore opt for a *safe termination* strategy [28]. When no ROI can be associated to a tracked object for t_{term} frames, the corresponding tracklet is terminated (we set $t_{term} = 3$). As soon as the object emerges from the occlusion again, a new tracklet will be generated, which can be connected to the same object by a higher-level tracker [9, 28]. In order not to confuse the later evaluation, we do not use such an additional tracking stage in this paper, but only report the raw tracklet results.

5 Carried Item Detection

Through our tracking process, we accumulate and refine a 3D model of each tracked person, which we can use in order to analyze its shape and volume in more detail. In the following, we propose a procedure to detect carried items, such as backpacks and hand- or shopping bags, from this information. The basic idea behind our approach is to compare the online model to a learned statistical shape template of unencumbered pedestrians in order to detect deviations that cannot be explained by the variation in GCT volume during a gait cycle.

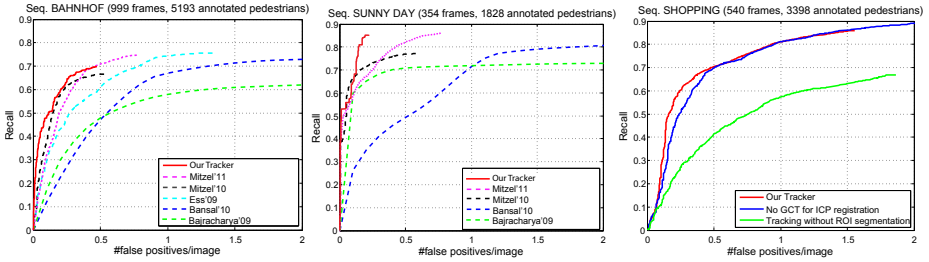


Fig. 6. Quantitative pedestrian tracking performance of our approach compared to different baselines on the BAHNHOF and SUNNY DAY sequences from [2] and on our own sequence SHOPPING. The results show that our approach outperforms the current state-of-the-art.

Pedestrian Model. In order to learn a statistical pedestrian shape model that matches the sensing characteristics of our noisy stereo input, we collected a training set of 12 GCT models of pedestrians moving in different directions over a duration of 15-20 frames from a separate training sequence. Half of the pedestrians were not carrying any bags or other items; the other half was carrying some item that we manually segmented out from the stereo depth images. We found this variability to be necessary in order to account for the different hand and arm positions people adopt when carrying items.

As the different training models capture different views of the tracked pedestrians, we used their trajectory directions in order to align the models to a consistent orientation. After this registration, we merged the distance distributions stored in the corresponding rays. The resulting full model is shown in Fig. 5(c). As can be expected, the variance of the distances in the bottom part is higher due to strong leg articulation.

Model Alignment. In order to compare distance distributions between the corresponding model and object rays, we first require an accurate model alignment. We therefore first register the tracked object GCT to the learned model GCT based on the observed walking direction. Before comparing the two models, however, an additional fine registration step is required. This is necessary, because the object center was only selected based on a rough guess when the track was initialized, potentially resulting in an assignment of surface points to different rays between the two models. We therefore first reconstruct a point cloud from the tracked object GCT and align this point cloud to the learned model by an additional ICP registration. We then shift the center of the object GCT and adapt the stored distance distributions accordingly. After this registration, the distance distributions are comparable.

We are interested in determining which parts of the tracked object surface cannot be explained by the natural statistical variations in body proportions, articulations, and shapes of pedestrians that are not carrying items. In order to do this, we compare the stored distance distributions of corresponding model rays using the Bhattacharyya distance $bhatta(P,Q) = 1 - \sqrt{\sum_i p_i q_i}$. Fig.5(d) shows a model of a tracked object with carried items after comparison to the learned model. As can be seen, the bags are clearly separated from the the body.

Carried Item Segmentation. The final carried item segmentation is based on Conditional Random Fields (CRFs), which have been extensively used for image

Fig. 7. Example tracking results on the test sequence SHOPPING

segmentation. We use two labels $L = \{\text{carried item}, \text{no carried item}\}$ and the following energy function, defined over unary potentials $\psi_i(y_i)$ and pairwise potentials $\psi_{ij}(y_i, y_j)$:

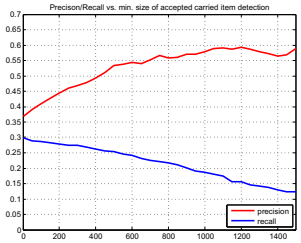
$$\mathbf{E}(\mathbf{y}) = \sum_{i \in V} \psi_i(y_i) + \sum_{(i,j) \in E} \psi_{ij}(y_i, y_j) \tag{3}$$

We define the unary potentials on the pixel level as $\psi_i(y_i) = -log(p(y_i|r_i))$ where $p(y_i|r_i) = bhatta(r_{i,object}, r_{i,model})s(r_{i,object})$. Here, $bhatta(\cdot)$ is the Bhattacharyya distance between the distance histograms of the online tracked and learned model rays. $s(\cdot)$ is a sigmoid weighting function applied to the component of the ray distance $r_{i,object}$ that is orthogonal to the walking direction. The sigmoid function can be understood as a prior that carried items are usually not inside the leg area. Thus, the sigmoid keeps the weights of points that are further away from the object center and are likely to belong to the carried items, but suppresses the noise that comes from strong articulations in the leg area. The unary potentials are set for the pixel positions of the back-projected rays. For the pairwise potentials, we use a contrast-sensitive Potts model [29] based on the image colors as $\psi_{ij}(y_i, y_j) = \theta_{i,j} \exp(-\beta||x_i - x_j)||^2)\delta(y_i \neq y_j)$. We solve the CRF using standard Graph-Cuts with α-expansion [30].

6 Experimental Results

Datasets. To evaluate our approach, we report tracking performance on two popular publicly available datasets: BAHNHOF and SUNNY DAY from [2]. Both sequences were captured from a stereo camera setup which was mounted on a child stroller. The BAHNHOF sequence consists of 999 frames with 5193 pedestrian annotations. SUNNY DAY contains 354 frames with 1867 annotated pedestrians. In addition, we report quantitative and qualitative performance for several additional own sequences, which we captured in busy shopping streets using a similar child stroller setup. Our sequences contain more complex scenarios with many unknown objects, such as child strollers, wheelchairs, suitcases, and walking aids. We annotated one of those sequences (SHOPPING) in a similar fashion as the above, resulting in 3398 pedestrians annotations over 540 frames. We will make all sequences publicly available at `http://www.vision.rwth-aachen.de/projects/gct`, including annotations, stereo depth, visual odometry, and estimated ground planes. We hope that this will allow other researchers to build upon our results even without having to assemble a complete system on their own.

For stereo range estimation we used the robust publicly available approach proposed by [31]. Pedestrian classification was done using the DPM detector from [8].



Object types	mostly tracked ($> 80\%$)	nearly tracked ($> 50\%$)	not tracked ($< 20\%$)
Child stroller	5	0	0
Walking aid	0	1	0
Suitcase	1	1	0
Wheelchair	2	0	0
Garbage bin	6	1	0
Bicycle	2	3	4
Advertising rack	0	2	3
Ticket dispenser	1	1	0

Fig. 8. (left) Per-pixel evaluation of the carried item segmentation accuracy. The carried items where annotated in every 4^{th} frame of the BAHNHOF sequence, resulting in a total of 282 annotated items over 250 frames. (right) Performance evaluation for tracking of unknown objects.

Pedestrian Tracking Performance. We first verify that our approach achieves competitive tracking performance for pedestrians. For this, we use the evaluation criteria presented by [2]. In every frame we measure the intersection-over-union of tracked person bounding boxes and annotations. Detections with an overlap larger than 0.5 are accepted as correct. The results are presented in terms of recall *vs*. false positives per image (fppi). Figs. 6(a) and (b) present the achieved performance for BAHNHOF and SUNNY DAY. It can be seen that our approach achieves state-of-the-art performance on BAHNHOF and significantly outperforms previously published results on SUNNY DAY. Note that, in contrast to tracking-by-detection approaches where the detector needs to be executed for the entire image in each frame, we only evaluate the detector for 0.7 ROIs per frame on average.

In addition, Fig. 6(c) shows the tracking performance on the SHOPPING sequence (red curve). As can be seen, even in such a complex and highly crowded scenario the achieved performance is comparable to the ones observed for the BAHNHOF and SUNNY DAY sequences. We additionally compare our approach to two baselines. The green curve in Fig. 6(c) illustrates the performance of our tracking system without the ROI segmentation and association step in each frame. Instead, we only perform an initial segmentation for each object in the first frame where it is visible and then sample 3D points for the ICP registration step in an uncertainty region around the projected position in all further frames (similar to [20]). This works reasonably well for individual objects. However, it causes drifting of the tracks for pedestrians moving in a group. The blue curve in Fig. 6(c) shows the performance of our system when using standard ICP based on the original 3D points from the previous frame (instead of model-based ICP based on the accumulated GCTs) for the registration with the new points of the associated ROI. As can be seen, using the GCT model for tracking results in more robust performance, which can be explained by a more accurate object representation that compensates for noisy and fragmentary stereo data. Fig 7 shows some qualitative pedestrian and unknown object tracking results on the SHOPPING sequence. It can be seen that our system is able to track most of the visible persons and objects correctly.

Object Tracking Performance. For assessing the performance of our approach for tracking unknown objects, we have processed 6060 frames of video material, which contained the unknown objects listed in Tab. 8. As can be seen, important dynamic objects such as child strollers and wheelchairs are correctly tracked most of the time. The performance for some other object types, such as stationed bicycles, is still lower

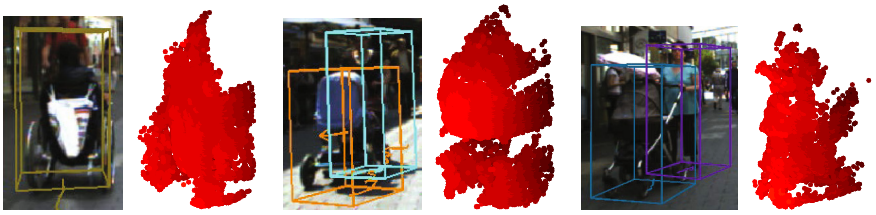


Fig. 9. Tracking results and accumulated GCT models for several unknown objects

Fig. 10. Carried item detection results (in red). The bottom row shows some failure cases which are caused by depth smoothing, excessive body proportions, and unusually wide clothing.

due to the fact that the bicycles are often segmented in two or more areas. For each tracked object, we accumulate a GCT model capturing its detailed shape. Some of the resulting shape models are shown in Fig. 9, and additional tracking results are shown in Fig. 11. As can be seen, our approach succeeds in capturing the shape essentials, providing sufficient detail to support future recognition approaches.

Carried Item Detection Performance. For evaluating the performance of the carried item detection we labeled all carried items (in total 282) in every fourth frame of the BAHNHOF sequence. We compared our hypothesized segmentation results pixel-wise with the labeled data, resulting in the precision and recall plots in Fig. 8(left). Removing small noisy detections results in higher precision at the cost of some recall. Overall, our approach achieves a carried item segmentation precision of about 60%. Given the very noisy nature of our stereo input data and the small size of the items in question, we think this is a very encouraging result. Some example results are shown in Fig. 10.

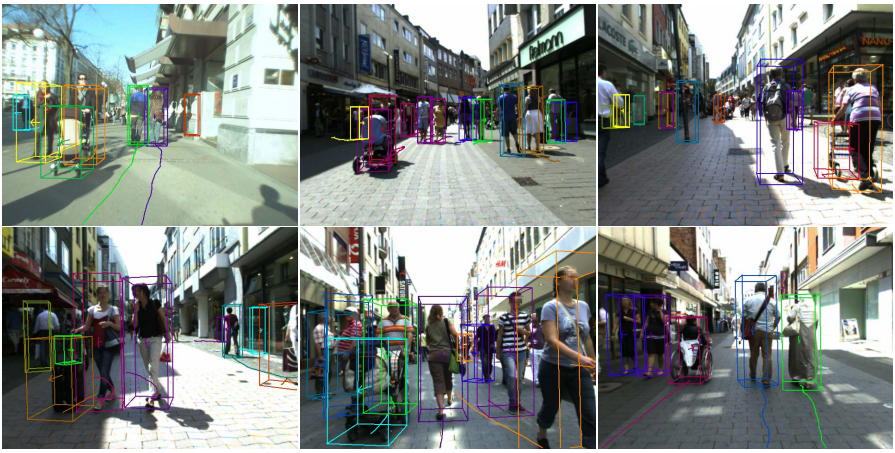


Fig. 11. Results for tracking of pedestrians and unknown objects

7 Conclusion

We have presented a novel 3D tracking approach for tracking both known and unknown objects in busy scenes based on stereo range data. The core of our approach is a novel 3D representation for objects which is built online by accumulating object surface information over the tracking process based on ICP registration. Apart from achieving state-of-the-art tracking performance, the adaptive 3D model representation allows us also to analyze the shape of tracked persons in more detail. By comparing the observed shape with a learned probabilistic shape template of pedestrians, we can hypothesize the parts of the shape that are likely to be carried items. Additionally, we can not only track a large number of pedestrians and unknown objects over time, but we save computation, since the person classifier needs to be applied only once for each object track.

Acknowledgments. This project has been funded, in parts, by the cluster of excellence UMIC (DFG EXC 89).

References

1. Leibe, B., Schindler, K., Van Gool, L.: Coupled Object Detection and Tracking from Static Cameras and Moving Vehicles. PAMI 30(10), 1683–1698 (2008)
2. Ess, A., Leibe, B., Schindler, K., Van Gool, L.: Robust Multi-Person Tracking from a Mobile Platform. PAMI 31(10), 1831–1846 (2009)
3. Bajracharya, M., Moghaddam, B., Howard, A., Brennan, S., Matthies, L.: A Fast Stereo-based System for Detecting and Tracking Pedestrians from a Moving Vehicle. IJRS 28(11-12), 1466–1485 (2009)
4. Bansal, M., Jung, S.H., Matei, B., Eledath, J., Sawhney, H.S.: A real-time pedestrian detection system based on structure and appearance classification. In: ICRA (2010)
5. Kuo, C.H., Huang, C., Nevatia, R.: Multi-Target Tracking by On-Line Learned Discriminative Appearance Models. In: CVPR (2010)
6. Andriluka, M., Roth, S., Schiele, B.: Monocular 3D Pose Estimation and Tracking by Detection. In: CVPR (2010)

7. Wojek, C., Walk, S., Roth, S., Schiele, B.: Monocular 3D Scene Understanding with Explicit Occlusion Reasoning. In: CVPR (2011)
8. Felzenszwalb, P., Girshick, B., McAllester, D., Ramanan, D.: Object Detection with Discriminatively Trained Part-Based Models. PAMI 32(9), 1627–1645 (2010)
9. Mitzel, D., Horbert, E., Ess, A., Leibe, B.: Multi-person Tracking with Sparse Detection and Continuous Segmentation. In: Daniilidis, K., Maragos, P., Paragios, N. (eds.) ECCV 2010, Part I. LNCS, vol. 6311, pp. 397–410. Springer, Heidelberg (2010)
10. Besl, P.J., Mckay, H.D.: A method for registration of 3-D shapes. PAMI 14(2) (1992)
11. Luber, M., Spinello, L., Arras, K.: People Tracking in RGB-D Data With Online-Boosted Target Models. In: IROS (2011)
12. Prengaman, R., Thurber, R., Bath, W.: Retrospective Detection Algorithm for Extraction of Weak Targets in Clutter and Interference Environments. In: IEEE Int. Radar Conf. (1982)
13. Salmond, D., Birch, H.: A Particle Filter for Track-Before-Detect. In: American Control Conf. (2001)
14. Danescu, R., Oniga, F., Nedevschi, S.: Modeling and Tracking the Driving Environment With a Particle-Based Occupancy Grid. IEEE Trans. Intel. Transp. Syst. 12(4), 1331–1342 (2011)
15. Geronimo, D., Lopez, A., Sappa, A., Graf, T.: Survey of Pedestrian Detection for Advanced Driver Assistance Systems. PAMI 32(7), 1239–1258 (2010)
16. Petrovskaya, A., Thrun, S.: Model Based Vehicle Detection and Tracking for Autonomous Urban Driving. AR 26(2–3), 123–139 (2009)
17. Kaestner, R., Maye, J., Siegwart, R.: Generative Object Detection and Tracking in 3D Range Data. In: ICRA (2012)
18. Gavrila, D., Munder, S.: Multi-Cue Pedestrian Detection and Tracking from a Moving Vehicle. IJCV 73(1), 41–59 (2007)
19. Keller, C., Fernandez-Llorca, D., Gavrila, D.: Dense Stereo-Based ROI Generation for Pedestrian Detection. In: Denzler, J., Notni, G., Süße, H. (eds.) Pattern Recognition. LNCS, vol. 5748, pp. 81–90. Springer, Heidelberg (2009)
20. Mitzel, D., Leibe, B.: Real-Time Multi-Person Tracking with Detector Assisted Structure Propagation. In: ICCV CORP Workshop (2011)
21. Feldman, A., Hybinette, M., Balch, T., Cavallaro, R.: The Multi-ICP Tracker: An Online Algorithm for Tracking Multiple Interacting Targets. J. Field Robotics (2012)
22. Teichman, A., Thrun, S.: Tracking-Based Semi-Supervised Learning. In: RSS (2011)
23. Teichman, A., Levinson, J., Thrun, S.: Towards 3D Object Recognition via Classification of Arbitrary Object Tracks. In: ICRA (2011)
24. Bjorkman, M., Kragic, D.: Active 3D Segmentation and Detection of Unknown Objects. In: IROS (2010)
25. Damen, D., Hogg, D.: Detecting Carried Objects in Short Video Sequences. In: Forsyth, D., Torr, P., Zisserman, A. (eds.) ECCV 2008, Part III. LNCS, vol. 5304, pp. 154–167. Springer, Heidelberg (2008)
26. Nistér, D., Naroditsky, O., Bergen, J.: Visual odometry. In: CVPR (2004)
27. Vedaldi, A., Soatto, S.: Quick Shift and Kernel Methods for Mode Seeking. In: Forsyth, D., Torr, P., Zisserman, A. (eds.) ECCV 2008, Part IV. LNCS, vol. 5305, pp. 705–718. Springer, Heidelberg (2008)
28. Kaucic, R., Perera, A., Brooksby, G., Kaufhold, J., Hoogs, A.: A Unified Framework for Tracking through Occlusions and Across Sensor Gaps. In: CVPR (2005)
29. Rother, C., Kolmogorov, V., Blake, A.: Grabcut: Interactive Foreground Extraction Using Iterated Graph Cuts. In: SIGGRAPH (2004)
30. Boykov, Y., Veksler, O., Zabih, R.: Fast Approximate Energy Minimization via Graph Cuts. PAMI 23(11), 1222–1239 (2001)
31. Geiger, A., Roser, M., Urtasun, R.: Efficient Large-Scale Stereo Matching. In: Kimmel, R., Klette, R., Sugimoto, A. (eds.) ACCV 2010, Part I. LNCS, vol. 6492, pp. 25–38. Springer, Heidelberg (2011)