2014 IEEE International Conference on Robotics & Automation (ICRA)
Hong Kong Convention and Exhibition Center
May 31 - June 7, 2014. Hong Kong, China

Real-Time RGB-D based People Detection and Tracking for Mobile Robots and Head-Worn Cameras

Omid Hosseini Jafari[1], Dennis Mitzel[1], Bastian Leibe[1]

***Abstract*— We present a real-time RGB-D based multi-person detection and tracking system suitable for mobile robots and head-worn cameras. Our approach combines RGB-D visual odometry estimation, region-of-interest processing, ground plane estimation, pedestrian detection, and multi-hypothesis tracking components into a robust vision system that runs at more than 20fps on a laptop. As object detection is the most expensive component in any such integration, we invest significant effort into taking maximum advantage of the available depth information. In particular, we propose to use two different detectors for different distance ranges. For the close range (up to 5-7m), we present an extremely fast depth-based upper-body detector that allows video-rate system performance on a single CPU core when applied to Kinect sensors. In order to cover also farther distance ranges, we optionally add an appearance-based full-body HOG detector (running on the GPU) that exploits scene geometry to restrict the search space. Our approach can work with both Kinect RGB-D input for indoor settings and with stereo depth input for outdoor scenarios. We quantitatively evaluate our approach on challenging indoor and outdoor sequences and show state-of-the-art performance in a large variety of settings. Our code is publicly available.**

I. INTRODUCTION

In this paper, we address the problem of RGB-D based people detection and tracking from the perspective of a moving observer. We focus on two scenarios: a mobile robot navigating through busy urban environments [1] or indoor shopping zones and a moving human wearing a head-mounted camera system for future Augmented Reality applications (Fig. 1). Those scenarios pose specific challenges that make robust people tracking difficult. In busy environments, many people close to the cameras are only partially visible in the camera view due to occlusions at the image boundaries. This is a particular challenge for mobile robotics applications, since people close to the robot are also the most important ones to track for dynamic obstacle avoidance. At the same time, those cases are not well-handled by state-of-the-art object detectors [2], [3], which often fail under occlusion. This problem is even more severe for the head-mounted scenario, where close-by people additionally undergo perspective distortion due to the elevated viewpoint.

A second major issue is computational cost. In order to be useful for robot navigation decisions and mobile AR, people tracking needs to operate in real-time and with as little computation as possible. Although real-time detection approaches have already been proposed in the vision community [4], [5], [6], they often build upon GPU processing, which implies significant energy consumption and thus reduced robot autonomy. We argue that object detection should take advantage of all available scene information in order to reduce the effort such detectors have to perform in the context of a robotic application.

[1]Computer Vision Group, RWTH Aachen University, Germany
surname@vision.rwth-aachen.de

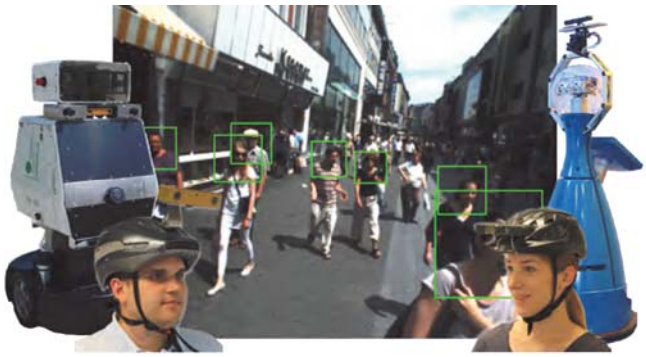


Fig. 1. Our people detection and tracking approach is suitable for mobile robots and head-worn cameras and can work with both stereo data for outdoor settings (left) and with Kinect RGB-D input for indoor scenarios (right). At its core is an extremely fast depth-based upper-body detector that allows robust performance in busy settings (middle).

In this work, we therefore aim at taking maximum advantage of the available depth information from RGB-D sensors in order to reduce the object detection effort. We consider two such sensor configurations, stereo depth for outdoor scenarios and Kinect RGB-D data for indoor settings, and develop an approach that can work with both types of input. The key idea of our approach is to use the depth information for region-of-interest (ROI) extraction and very fast people detection in the close range, where depth measurements are reliable, while simultaneously extrapolating scene geometry information to constrain the search space for appearance-based object detection in the far range. We construct a robust and efficient ROI processing pipeline that performs those steps together with visual odometry estimation and multi-hypothesis tracking. Based on Kinect RGB-D input, our system runs at video frame rate on a single CPU core (no GPU involved) when only considering close-range detections and still at 18 fps (including GPU processing) when adding the far-range detector. We have fully integrated our code into ROS and make the code publicly available for research purposes.

II. RELATED WORK

Multi-object detection and tracking from a mobile platform is a core capability for many applications in mobile robotics and autonomous vehicles. Several frameworks have been proposed addressing this task. The approach by [7] uses

an ensemble of detectors based on multiple cues and fuses the detector output in an RJ-MCMC tracker. [8] propose a tracking framework which first places the detection bounding boxes in a 3D space-time volume using visual odometry and a ground plane and then generates trajectory hypotheses by linking the detections over frames with Extended Kalman Filters. To obtain a final set of trajectory hypotheses they perform model selection in an MDL framework. However, both approaches are computationally very expensive and do not reach real-time speeds.

There are several of state-of-the-art object detectors such as [2], [3] which yield remarkably accurate detections for fully observed pedestrians. However, in our scenarios where people are often occluded by image borders these detectors yield poor performance. Furthermore, these approaches are computationally expensive, requiring dedicated optimization efforts to make them applicable to mobile platforms [9].

For detecting pedestrians that are undergoing partial occlusions, [10] propose to combine a full object detector and multiple object part detectors in a mixture of experts based on their expected visibility. [11] use stereo and flow cues in order to support learned local body part detectors which are combined in a mixture-of-experts framework. For addressing the occlusion problem [12] propose using a modified SVM framework which combines HOG and LBP features. However, all of these approaches require high computational effort for feature extraction or classifier evaluation.

[13] propose a multiresolution model that employs DPM for detecting pedestrians that are close to the camera and a rigid template for finding small instances of objects. The detections are re-scored using geometrical constraints, such as that pedestrians are standing or moving on a ground plane. We use a similar strategy when applying two different detectors for different distance ranges.

In order to reduce the computational effort for object detection several approaches have been proposed to restrict the execution of the detector to only few ROIs that are extracted based on, *e.g.*, stereo range data [14], [15], [16], [17], motion [18] and scene geometry [19], [6], [20]. We use a similar strategy of ROI processing based on stereo information in order to reduce the search space for our close-range upper-body detector. In addition, we exploit scene geometry for the far-range full-body detector. As a result, we reduce the computational effort and also the number of possible false positives, since only image regions are evaluated that are likely to contain target objects.

Several existing approaches incorporate the stereo information in order to improve detection performance. [16] employ depth cues for first extracting the ROIs. The detection hypotheses are generated by measuring the Chamfer distance between a learned shape contour model and the image input. [21] propose a full-body pedestrian detector using dense depth data from an RGB-D Kinect camera. Based on the idea of HOG, they introduce Histograms of Oriented Depths as a new feature that follows the same computation procedure as HOG. The above-mentioned approach by [7] uses an ensemble of detectors where one of the detectors is a binary depth template that is used to measure the distance to an observed shape of a human. However, as already shown in [17], using a continuous normalized-depth template yields significantly better performance, which we further improve using extensions proposed in this paper.

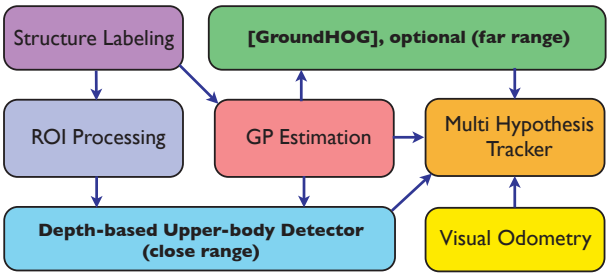


Fig. 2. Overview of our full detection and tracking framework.

III. OVERVIEW

Fig. 2 shows a compact overview of our proposed detection and tracking system. For each new RGB-D frame, we first perform a *structure labeling* to classify the 3D points into three different classes (`object`, `ground plane`, `fixed structure`). `Fixed structure` points are directly filtered out. Points that are classified as `ground plane` are passed to the ground plane estimation module, which fits a plane to these points using RANSAC. Points that belong to the `object` class are passed to the ROI processing module. This module extracts ROIs by projecting the 3D points onto the ground plane and segmenting the resulting blobs into individual objects. For each extracted 3D ROI, we generate a corresponding ROI in the image plane through backprojection. The 2D ROIs are passed to the depth-based upper-body detector, which slides a learned upper-body template over the ROIs and computes a distance matrix consisting of the Euclidean distances between the template and each overlayed normalized depth image segment. The upper-body detector operates on depth only and consequently is limited to the range available from depth sensors (*e.g.*, Kinect), which is usually up to 5 meters. In order to obtain detections also for pedestrians at farther range, we use *groundHOG*, a GPU-optimized detector proposed by [6]. This detector is based on HOG features and allows us to use the estimated scene geometry (ground plane) to reduce the search region in the image to the minimal region that can contain geometrically valid detections. Finally, we estimate the camera motion in the *visual odometry* component. We then use this motion, together with the ground plane and the detections from both detectors, in the *tracking* module, where the bounding boxes are converted to ground plane coordinates and are associated into trajectories using Extended Kalman Filters (EKFs) and a multi-hypothesis handling scheme. The full system, including all the components visualized in Fig. 2 except for *groundHOG*, runs at 24fps on a single CPU of a gaming notebook (Intel i7-3630QM, 12GB RAM, NVIDIA GeForce GT650m). When including *groundHOG*, an NVIDIA GPU is required and we can reach 18fps on the same gaming laptop. All the modules are ported to

(a) (b) (c)

Fig. 3. Visualization of our elevated structures classification. (a) Height histogram. (b) Labeling of the 3D points according to the different height bands (**green** - ground plane band, **red** - object band, **yellow** - free space band, **blue** - elevated structure band). (c) Final classification result of points into three classes.

ROS and are made publicly available on our group webpage (`http://www.vision.rwth-aachen.de/software`).

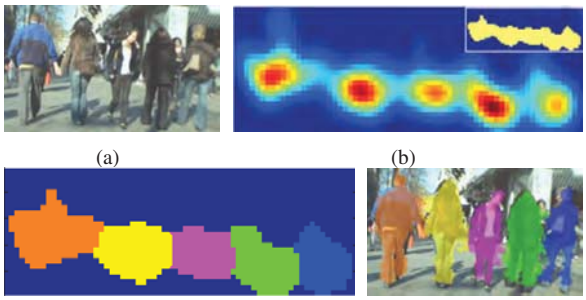
IV. ROI PROCESSING

A. Point Cloud Labeling

Several approaches have been proposed for static obstacle detection in robotics [22], [23], where point clouds from 3D sensors are fused over a time window and the resulting 3D points are then projected to the ground plane forming an occupancy map. From the occupancy map, the free space where the robot can move can be computed using dynamic programing. We apply a similar approach for fusing the point clouds in a reference frame using the camera motion estimated with visual odometry (robot odometry could be used as well). These fused 3D points are then segmented into three different classes: `objects`, `ground plane` and `fixed structures`. Given this segmentation, we can exclude 3D points on `fixed structures` before ROI extraction, which will reduce the number of typical false pedestrian detections in the reflections of shopping windows.

Before the classification step, we first need to obtain an accurate ground plane estimate. To this end, we compute the occupancy map from the fused point clouds by projecting all 3D points within a 2m height corridor to a rough estimate of the ground plane based on the camera height of the recording vehicle. The 3D points inside bins with a high density are excluded and the majority of the remaining points corresponds to points on the real ground plane, which we estimate by plane fitting using RANSAC [24].

The classification pipeline is visualized in Fig. 3. The incoming point clouds are fused over a time period of 5-10 frames and the accumulated 3D points are projected to a 2D ground plane occupancy histogram. For each bin of the 2D histogram, we compute a height histogram from the corresponding points. The height histogram has four height bands: the `ground plane` band, `object` band, the `free corridor` band and the `elevated structures` band, Fig. 3(b). The free space corridor has a significant effect on classification performance. The simplest way to classify elevated structures would be to assign the points in bins with a high point density in the `elevated structures` band to the label `fixed` object. However, in shopping street scenarios this will often cause mislabeling of the objects due to overhanging building parts. With a free space corridor we make the assumption that there is always a free space (at 2m-2.3m) between the heads of pedestrians and overhanging parts from buildings or trees. This means that if for a bin in the 2D histogram the point density in the free space corridor is low (number of fused frames), the points in the object band are labeled as objects and the points inside the free space bin and the elevated bin are labeled as `fixed structures`. Even though we call the third band `free space` band, suggesting that this band should be free of points, we still make the assumption that due to noisy depth and visual odometry input some points fall inside. For bins with a high density (number of points higher than the number of fused frames) in the free space corridor band, all points are labeled as belonging to the class `fixed structures`. This results in a final point classification as shown in Fig. 3(c). As can be seen, we obtain an accurate labeling for all three classes. The 3D points labeled as `fixed structures` are excluded from further processing.



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

Fig. 4. Visualization of our stereo ROI extraction and segmentation procedure. (a) Original image. (b) Projection onto ground plane of the five persons. In the upper right corner is the corresponding connected component of the histogram. (c) Segmentation of the ROI into individual object clusters using the Quick Shift algorithm [25]. (d) Corresponding point cloud segmentation projected back into the image.

B. ROI Extraction

Following the approaches by [15], [14], [17], we project all 3D points labeled as `objects` onto the ground plane. We collect the points in a 2D histogram and weight them according to their quadratic distance to the camera in order to compensate for the fact that objects that are further away consist of fewer points and would therefore be rejected in further processing steps. The histogram bins (Fig. 4(b)) are first smoothed with a Gaussian Filter ($\sigma = 2.0$ in x-direction and $\sigma = 3.0$ in z-direction) and are then thresholded in order to remove irrelevant areas caused by noise. The remaining bins after thresholding are grouped into connected components using an 8-neighborhood. 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 the highest point within the corresponding bins, oriented parallel to the camera. By projecting this rectangle to the image, we obtain the corresponding image region that can then be evaluated by the detector. We already applied this ROI extraction mechanism successfully in our previous work [17], reducing not only the computational effort for the detector through the restricted search space, but also lowering the number of false positives.

C. ROI Segmentation

As can be seen in Fig. 4(b), people walking close to each other are often fused into one connected component in the ground projection. This means that, when applying our upper-body detector, we will have to rescale the template in order to compensate for different sizes of pedestrians. Using a multi-scaling approach then requires the use of a non-maximum suppression procedure, since we might obtain several detections around an actual correct detection coming from neighboring scales. In order to avoid rescaling, we propose to instead segment the connected components further into distinctive regions using a smoothed version of the original histogram (*c.f.* Fig. 4(c)). For segmentation, we employ the Quick Shift algorithm [25], which is a fast variant of Mean-Shift [26]. 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:

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

where P_i is the density estimate for point x_i with the kernel function θ, and $d(x_i, x_j)$ is the distance between points x_i and x_j. We start Quick Shift for each point in the ground projection histogram and link this point to its respective neighbor with the highest density until we reach a mode. The points on the way to the mode are automatically associated to this mode. The Quick Shift procedure yields a segmentation of the ROIs into individual objects, as shown in Fig. 4(c). Fig. 4(d) shows the corresponding segmented point cloud.

V. PEDESTRIAN DETECTION

A. Depth based Upper-Body Detector (Close Range)

The detection module for pedestrians at close range is based on our depth-based upper-body detector [17], which we improved significantly by introducing several novel extensions. Before presenting the extensions, we will briefly recap the original detector idea (*c.f.* Fig 5). The core of the detector is a normalized-depth template which is generated from annotations by averaging the bounding box content. In each frame, this template is then slid over ROIs and is compared to the overlayed area using the Euclidean distance. The initial template scale is estimated based on the height of the 2D ROI (determined by the tallest person in the group). Since the ROIs may contain several close-by pedestrians, the template is rescaled in order to compensate for different person sizes. Due to this multi-scaling procedure, several positive detections can be generated around an actual pedestrian position, which are pruned to a final detection with a non-maximum-suppression (NMS) procedure.

The sliding window procedure, as applied in the original detector, still requires the classification function to be evaluated over a large set of candidate sub-windows. One possible speed-up could be to increase the position stride, which will reduce the number of classifier evaluations, but can cause imprecise object localization and increase the number of false negatives. Instead, we fix template evaluation only to local maxima of the contour, extracted from the corresponding ROIs as shown in Fig. 6(c). The contour is represented by a maximum y value at each x position, which is extracted from the corresponding depth image of an ROI (*c.f.* Fig. 6(b)). In order to reduce the number of local maxima in the depth image that are caused by noise, the contour is smoothed with a Gaussian before maxima extraction. As can be seen in Fig. 6(c), the extracted local maxima do not correspond to the exact location of the middle of the human head. For this reason the depth template is evaluated for each scale not only at the local maxima, but also in some neighboring region around them. In our experiments, we show that this approach significantly outperforms the original detector when using the local maxima extension. Furthermore, it reduces the computational time from 30ms to 24ms ($\approx$ 41fps on a single CPU of an Intel i7-3630QM, 12GB RAM), including ROI candidate generation, object detection, and tracking.

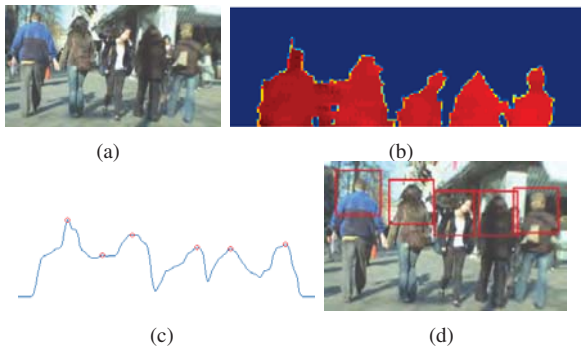
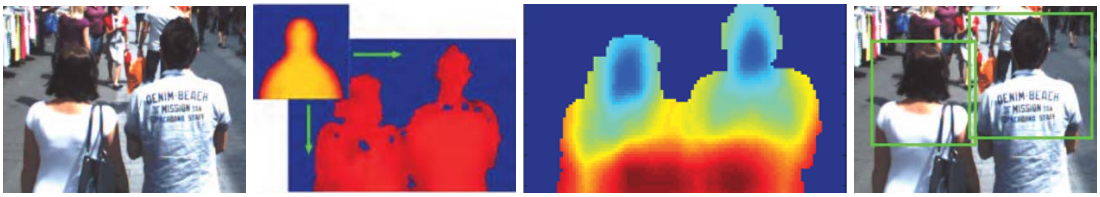


Fig. 6. Overview of the contour based approach for upper body detection using local maxima. (a) Original image. (b) Corresponding depth image crop. (c) Resulting contour with the extracted local maxima. (d) Resulting detections.

As mentioned before, using the local maxima in the extracted ROI depth image in order to find a correct scale of the pedestrian requires rescaling the template. However, using the ROI segmentation yields ROIs that contain only one person which allows us to fix the scale of the template and to apply it only once. So, by combining local maxima with the ROI segmentation we get better detection performance than using local maxima on original ROIs, while further reducing the computational time from 24ms to 23ms per frame ($\approx$ 43fps).

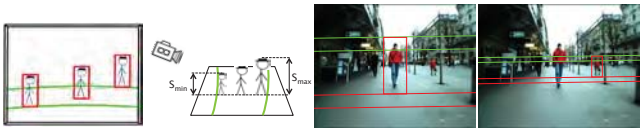
B. Ground HOG (Far Range)

For the far-range pedestrian detector we employ the GPU based full-body HOG detector proposed by [6]. In contrast to a classical sliding window HOG detector, as originally proposed by [2], [6] exploit the estimated scene geometry in order to restrict the search space of the detector. Assuming that pedestrians stand on the ground plane and have a certain height range $\in [s_{min}, s_{max}]$ (*c.f.* Fig. 7(a)), [6] derive a closed-form expression for the minimal range of pixels at each detector scale that need to be evaluated in order to only return geometrically valid detections (*c.f.* Fig. 7(b-c)). Since the ROIs can be computed very efficiently for each image scale and since they significantly reduce the



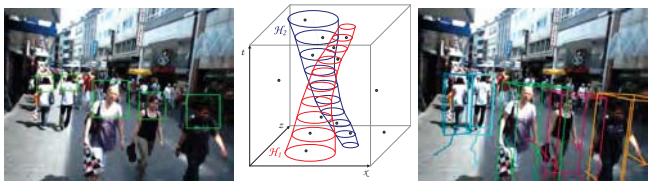
(a) (b) (c) (d)

Fig. 5. Depth based detection pipeline. (a) Backprojected 3D ROI. (b) Apply depth template in sliding window manner. (c) Corresponding distance matrix. (d) Final detections after Non-Maximum-Suppression.



(a) (b) (c)

Fig. 7. Visualization of the *groundHOG* approach. (a) By assuming that pedestrians stand on a ground plane and have a known height range, *groundHOG* derives a closed-form expression for the minimal search corridors at each detection scale that can contain geometrically valid detections (b-c).



(a) (b) (c)

Fig. 8. Multi-hypothesis tracking pipeline. (a) Object detections coming from the upper body detector (b) Hypothesis Generation process where the detections are linked together using a bi-directional Extended Kalman Filter under constraints of motion and appearance. (c) Hypothesis Selection.

image space where the feature computation needs to be performed, we reach a factor 2-4 speed-up (on top of all code-level optimizations) compared to a classical sliding window approach. In numbers, this means that we achieve a run-time of 40ms (25fps) for all scales on our gaming laptop. However, since in our full system our depth based upper-body detector reaches remarkable performance for close-range pedestrians already (up to 7m distance), we apply *groundHOG* only to farther ranges starting from 7m. This means we significantly reduce the number of scales that need to be evaluated, further reducing the computational effort on the GPU down to 30ms per frame (33fps).

VI. PEDESTRIAN TRACKING

In the tracking component we fit a set of trajectories to the 3D pedestrian locations detected by our upper-body detector and by *groundHOG*. The goal is to select the subset of trajectory hypotheses that best explains the observed evidence. We use an extended version of the robust multi-person tracking framework proposed by [27]. Briefly stated, the tracking framework executes the following two steps. It first generates candidate trajectories through trajectory growing (linking new detections to existing space-time trajectories using a bi-directional EKF) (*c.f.* Fig. 8(b)) and by starting new trajectories that run backwards in time for each new detection (in order to recover previously lost tracks based on the new evidence). In both cases, the EKF is guided by a pedestrian-specific motion and appearance model (constant-velocity). The trajectory growing process generates an over-complete set of trajectories, which are then pruned to an optimal subset using hypothesis selection. For this step, we employ the MDL model selection framework by [27], which amounts to solving a quadratic boolean problem. As shown in our experiments, the tracking framework significantly improves upon the detection performance by filtering out false positives and compensating for false negatives by interpolating with the EKF.

Average run-time in ms	
Visual Odometry	15
ROI processing	10
Ground plane estimation	4
Upper-body detector	7
groundHOG - GPU	14
Multi-hypothesis tracker	6
Total w/o groundHOG	42
Total w/ groundHOG	56

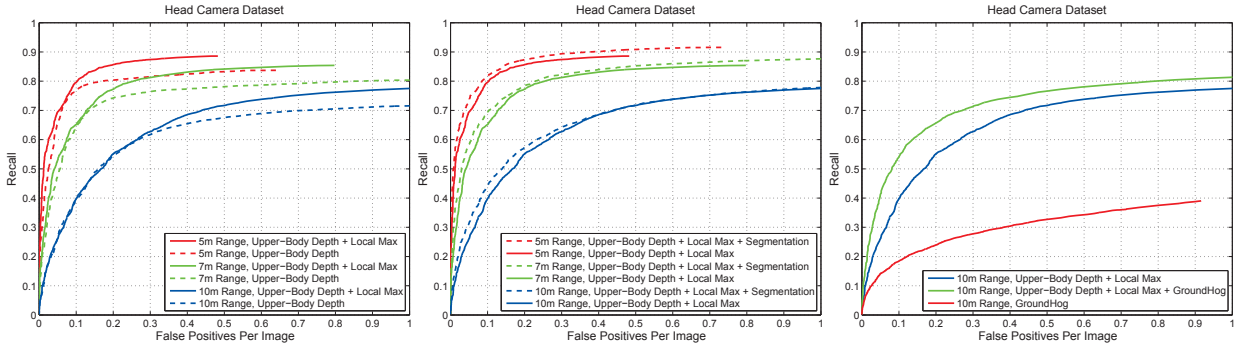
TABLE I

VII. EXPERIMENTAL RESULTS

We present detailed experimental evaluation results on three different sequences which were captured with three different camera setups.

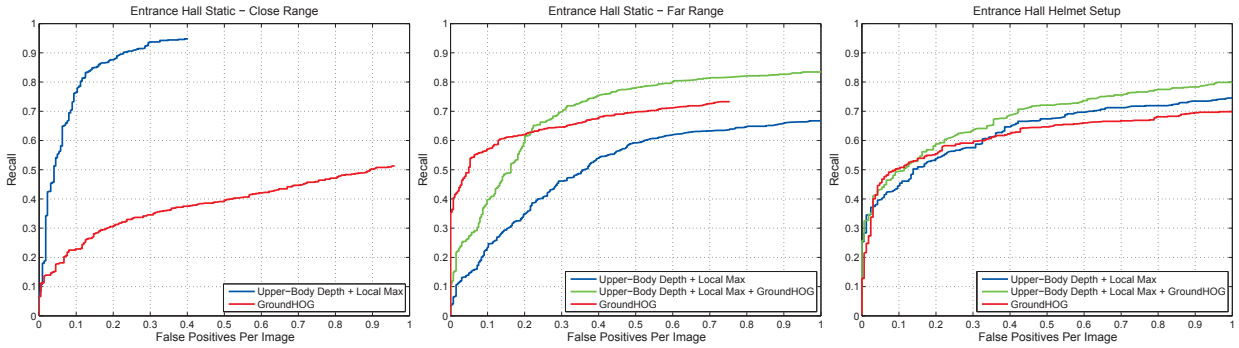
A. Datasets

The first dataset is the SUNNY DAY sequence from the Zurich Mobile Pedestrian corpus [22], which was captured outdoors with a stereo rig (14fps, 640×480 resolution) mounted on a child stroller that was moved through a busy inner-city pedestrian zone. SUNNY DAY consists of 999 frames, from which 354 are annotated with 1867 boxes around pedestrians. The second dataset was captured from a head-mounted camera setup (Bumblebee2) presented in [17] as HEAD MOUNTED CAMERA dataset. The images were captured at 15 fps in pedestrian shopping zones. The evaluation set consists of 2,543 frames with 19,461 pedestrian annotations. The challenge in this dataset is that many pedestrians are only partially visible due to strong occlusions at the image borders. In addition to those two existing datasets, we present a new dataset consisting of three sequences (in total 604 frames) captured indoors in an entrance hall of our university building. All three sequences were recorded using an Asus Xtion RGB-D sensor. Two of the sequences were captured using a static setup and contain many pedestrian passing by the camera. The third spequence was again captured with a helmet setup as in [17] and the person wearing the helmet was walking around



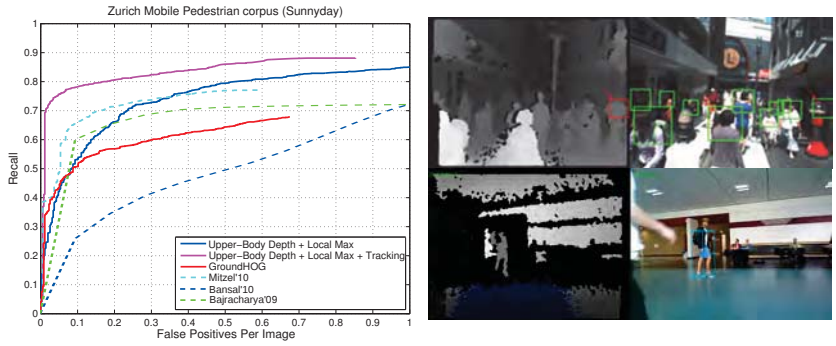
(a) (b) (c)

Fig. 9. Quantitative detection performance in recall vs. (fppi) on our HEAD MOUNTED CAMERA dataset presented in [17]. (a) Depth based upper detector [17] vs. local maxima extesion. (b) local maxima extension vs. combination of local maxima and ROI segmentation. (c) Comparing results of local maxima, combination of local maxima and groundHOG, and groundHOG.



(a) (b) (c)

Fig. 10. Quantitative detection performance in recall vs. (fppi) on three sequences of our new ENTRANCE HALL dataset. In all the sequences we compare the performance of only using groundHOG (for all scales), with the performance when using only our upper body detector and then also combination of both.



(a) (b)

Fig. 11. (a) Quantitative detection performance in recall vs. (fppi) on the SUNNY DAY sequence. (b) Illustration of typical failure cases of our approach: (top) false positives due to similar-looking background structures; (bottom) missing detection due to incomplete depth information outside the Asus Xtion sensing range.

groups of people. In all three sequences we have annotated 2,280 pedestrians with bounding boxes in a similar way as in the two previously described datasets.

B. Evaluation

For a quantitative evaluation, we measure bounding box overlap in each frame of the detected bounding boxes and the annotations and plot recall over false positives per image. Fig. 10 presents the performance on three sequences of our new ENTRANCE HALL dataset. Fig. 10(a) show the performance on the first sequence which contains pedestrians that are close to the camera (up to 4 meters). Since the depth sensors yield good depth for the close range, our upper body detector reaches excellent performance with 96% recall at already very low fppi rate of 0.4. However, the performance of groundHOG is significantly worse which is obvious since it is trained as full body pedestrian detector and it can not detect most of the pedestrians close to the camera. The situation changes if we look at the plot in Fig. 10(b) where we show the performance on the second sequence of the ENTRANCE dataset, where all pedestrians are fully visible and many of them are more than 6 meters away from the camera. Hence the *groundHOG* significantly outperforms our upper body detector, because for most of the pedestrians

we do not obtain any depth information from the depth sensor. However, when combining our upper body detector with *groundHOG* (upper body constrained for a distance up to 5 meters and *groundHOG* from 5-15m) we achieve a significantly better performance than *groundHOG* (*e.g.*, 9% higher recall at 0.5 fppi). Finally in the third sequence we obtained a similar performance for all three combinations (*c.f.* Fig. 10(c)), which can be explained by the fact that all pedestrians in this sequence are fully visible and still in the range where the depth sensor could provide depth information.

In Fig. 9(a-b) we evaluate the effect of the proposed extensions to our original depth based upper body detector [17] (denoted as upper-body depth detector in the plots). As can be seen in Fig. 9(a), adding the local maxima filtering step significantly improves detection performance for all distance ranges. When sliding the depth template over the entire extracted ROIs in the original approach, some false positives are found that can be avoided when restricting the evaluation to local maxima. Similarly, we obtain additional performance improvements when adding the ROI segmentation step (*c.f.* Fig. 9(b)), which helps split groups into ROIs for individual persons. The plot in Fig. 9(c) also confirms the observations we made in the evaluation on ENTRANCE HALL. We can significantly improve detection performance when combining our close-range upper-body detector with *groundHOG* for the far range. The HEAD CAMERA dataset contains many pedestrians that are occluded by the image borders, which explains why groundHOG fails here, but the upper body detector performs significantly better. By combining the two detectors, we can further boost performance by more than 10%.

Finally, we present evaluation results for the SUNNY DAY sequence in Fig. 11. In this sequence, most pedestrians are fully visible and all pedestrians are annotated. We run both the *groundHOG* detector and our upper-body detector here for the full distance range. An interesting observation is that our upper-body detector significantly outperforms *groundHOG* even for the far range. This can be explained by the different sensor setup (40cm baseline between the stereo cameras), which results in more precise depth measurements for up to 20m distance. Furthermore, we plot the performance when adding our tracking component (detection input from upper-body detector with local maxima filter), which further increases performance by filtering out false positives and interpolating trajectories during detection failures. The comparison to published baselines [14], [15], [28] shows that our approach reaches state-of-the-art tracking performance. Fig. 12 illustrates some qualitative detection and tracking results from different datasets. Some failure cases are shown in Fig. 11(b).

C. Computational Performance

Our current full system, including ROI processing (structure labeling and ground plane estimation), visual odometry, upper body object detection and tracking runs at 24fps, which is close to the frame rate delivered by an Asus Xtion RGB-D sensor. This performance was measured on a gaming laptop with Intel i7-3630QM, 12GB RAM, NVIDIA GeForce GT650m using our ENTRANCE HALL dataset. When including the GPU-based *groundHOG* detector, the overall frame rate drops to 18fps, but we can detect and track pedestrians in the far range as well, where the upper-body detector does no longer work due to missing depth information. Tab. I lists the timings for individual components of our full system. For the visual odometry we used a RGB-D based approach proposed by [29].

VIII. CONCLUSION

We have presented a fully integrated real-time detection and tracking system that achieves state-of-the-art performance. A main idea behind this integration was to take maximal advantage of the depth information readily available from current stereo and RGB-D sensors in order to simplify and speed up the detection process. For this purpose, we introduced a very fast depth-based upper-body detector operating on ROIs. Compared to previous approaches, we proposed several extensions to the ROI processing pipeline, which were shown to improve both run-time and detection performance. Supplementing the upper-body detector with an appearance-based full-body detector that performs significantly better at farther distance ranges increased the spatial operation radius of the tracking system to up to 15m, making it suitable for many mobile robotics tasks.

Acknowledgments: This work was funded, in parts, by ERC Starting Grant project CV-SUPER (ERC-2012-StG-307432) and EU projects STRANDS (ICT-2011-600623) and SPENCER (ICT-2011-600877).

REFERENCES

[1] R. Kuemmerle, M. Ruhnke, B. Steder, C. Stachniss, and W. Burgard, "A Navigation System for Robots Operating in Crowded Urban Environments," in *ICRA*, 2013.
[2] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," in *CVPR*, 2005.
[3] P. Felzenszwalb, B. Girshick, D. McAllester, and D. Ramanan, "Object Detection with Discriminatively Trained Part-Based Models," *PAMI*, vol. 32, no. 9, 2010.
[4] R. Benenson, M. Mathias, R. Timofte, and L. V. Gool, "Pedestrian Detection at 100 Frames per Second," in *CVPR*, 2012.
[5] V. Prisacariu and I. Reid, "fastHOG – a Real-Time GPU Implementation of HOG," Technical Report 2310/09, Dept. of Engineering Science, University of Oxford, 2009.
[6] P. Sudowe and B. Leibe, "Efficient Use of Geometric Constraints for Sliding-Window Object Detection in Video," in *ICVS*, 2011.
[7] W. Choi, C. Pantofaru, and S. Savarese, "A General Framework for Tracking Multiple People from a Moving Camera," *PAMI*, 2013.
[8] A. Ess, B. Leibe, K. Schindler, and L. V. Gool, "Object Detection and Tracking for Autonomous Navigation in Dynamic Environments," *IJRR*, vol. 29, no. 14, 2010.
[9] H. Cho, P. Rybski, A. Bar-Hillel, and W. Zhang, "Real-time Pedestrian Detection with Deformable Part Models," in *Intel. Vehicles Symp.*, 2012.
[10] C. Wojek, S. Walk, S. Roth, and B. Schiele, "Monocular 3D Scene Understanding with Explicit Occlusion Reasoning," in *CVPR*, 2011.
[11] M. Enzweiler, A. Eigenstetter, B. Schiele, and D. M. Gavrila, "Multi-Cue Pedestrian Classification with Partial Occlusion Handling," in *CVPR*, 2010.
[12] X. Wang, T. X. Han, and S. Yan, "An HOG-LBP Human Detector with Partial Occlusion Handling," in *ICCV*, 2009.
[13] D. Park, D. Ramanan, and C. Fowlkes, "Multiresolution models for object detection," in *ECCV*, 2010.

Fig. 12. Experimental results showing detection results for SUNNY DAY (1st row), ENTRANCE HALL (2nd row) and HEAD CAMERA (3rd row). In the last row, we show tracking results on the ENTRANCE HALL dataset.

[14] 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.

[15] 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*, vol. 28, no. 11-12, 2009.

[16] D. Gavrila and S. Munder, "Multi-Cue Pedestrian Detection and Tracking from a Moving Vehicle," *IJCV*, vol. 73, no. 1, 2007.

[17] D. Mitzel and B. Leibe, "Close-Range Human Detection for Head-Mounted Cameras," in *BMVC*, 2012.

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

[19] D. Geronimo, A. Sappa, D. Ponsa, and A. Lopez, "2D-3D-based On-Board Pedestrian Detection System," *CVIU*, vol. 114, 2010.

[20] D. Hoiem, A. Efros, and M. Hebert, "Putting Objects Into Perspective," in *CVPR*, 2006.

[21] L. Spinello and K. O. Arras, "People Detection in RGB-D Data." in *IROS*, 2011.

[22] A. Ess, K. Schindler, B. Leibe, and L. Van Gool, "Improved Multi-Person Tracking with Active Occlusion Handling," in *ICRA*, 2009.

[23] H. Badino, U. Franke, and R. Mester, "Free Space Computation using Stochastic Occupancy Grids and Dynamic Programming," in *ICCV Workshop on Dynamical Vision*, 2007.

[24] M. A. Fischler and R. C. Bolles, "Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography," *Commun. ACM*, vol. 24, no. 6, 1981.

[25] A. Vedaldi and S. Soatto, "Quick Shift and Kernel Methods for Mode Seeking," in *ECCV*, 2008.

[26] K. Fukunaga and L. Hostetler, "The Estimation of the Gradient of a Density Function, with Applications in Pattern Recognition," *IEEE Transactions on Information Theory*, vol. 21, no. 1, 1975.

[27] B. Leibe, K. Schindler, and L. Van Gool, "Coupled Object Detection and Tracking from Static Cameras and Moving Vehicles," *PAMI*, vol. 30, no. 10, 2008.

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

[29] A. S. Huang, A. Bachrach, P. Henry, M. Krainin, D. Maturana, D. Fox, and N. Roy, "Visual odometry and mapping for autonomous flight using an rgb-d camera," in *ISRR*, 2011.