Moving Obstacle Detection in Highly Dynamic Scenes

A. Ess[1], B. Leibe[2], K. Schindler[1,3], L. van Gool[1,4]

[1] Computer Vision Laboratory, ETH Zurich, Switzerland
[2] UMIC Research Centre, RWTH Aachen, Germany
[3] Computer Science Dept., TU Darmstadt, Germany
[4] ESAT/PSI-VISICS IBBT, KU Leuven, Belgium

{aess|leibe|schindler|vangool}@vision.ee.ethz.ch

***Abstract*— We address the problem of vision-based multi-person tracking in busy pedestrian zones using a stereo rig mounted on a mobile platform. Specifically, we are interested in the application of such a system for supporting path planning algorithms in the avoidance of dynamic obstacles. The complexity of the problem calls for an integrated solution, which extracts as much visual information as possible and combines it through cognitive feedback. We propose such an approach, which jointly estimates camera position, stereo depth, object detections, and trajectories based only on visual information. The interplay between these components is represented in a graphical model. For each frame, we first estimate the ground surface together with a set of object detections. Based on these results, we then address object interactions and estimate trajectories. Finally, we employ the tracking results to predict future motion for dynamic objects and fuse this information with a static occupancy map estimated from dense stereo. The approach is experimentally evaluated on several long and challenging video sequences from busy inner-city locations recorded with different mobile setups. The results show that the proposed integration makes stable tracking and motion prediction possible, and thereby enables path planning in complex and highly dynamic scenes.**

I. Introduction

For reliable autonomous navigation, a robot or car requires appropriate information about both its static and dynamic environment. While remarkable successes have been achieved in relatively clean highway traffic situations [3] and other largely pedestrian-free scenarios such as the DARPA Urban Challenge [6], highly dynamic situations in busy city centers still pose considerable challenges for state-of-the-art approaches.

For successful path planning in such scenarios where multiple independent motions and frequent partial occlusions abound, it is vital to extract semantic information about individual scene objects. Consider for example the scene depicted in the top left corner of Fig. 1. When just using depth information from stereo or LIDAR, an occupancy map would suggest little free space for driving (bottom left). However, as can be seen in the top right image (taken one second later), the pedestrians free up their occupied space soon after, which would thus allow a robotic platform to pass through without unnecessary and possibly expensive re-planning. The difficulty is to correctly assess such situations in complex real-world settings, detect each individual scene object, predict its motion, and infer a dynamic obstacle map from the estimation results (bottom right). This task is made challenging by the extreme degree of clutter, appearance variability, abrupt motion changes, and the large number of independent actors in such scenarios.

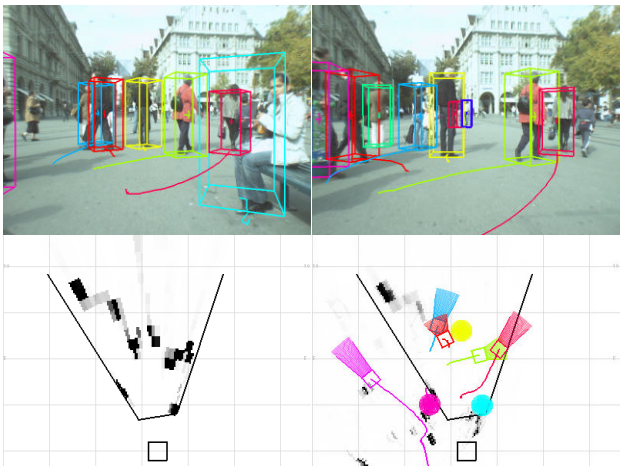

Fig. 1. A static occupancy map (bottom left) can erroneously suggest no free space for navigation, even though space is actually freed up a second later (top right). By using the semantic information from an appearance-based multi-person tracker, we can cast predictions about each tracked person's future motion. The resulting dynamic obstacle map (bottom right) correctly shows sufficient free space, as the persons walk on along their paths.

In this paper, we propose a purely vision-based approach to address this task. Our proposed system uses as input the video streams from a synchronized, forward-looking camera pair. To analyze this data, the system combines visual object detection and tracking capabilities with continuous self-localization by visual odometry and with 3D mapping based on stereo depth. Its results can be used directly as additional input for existing path planning algorithms to support dynamic obstacles. Key steps of our approach are the use of a state-of-the-art object recognition approach for identifying an obstacle's category, as well as the reliance on a robust multi-hypothesis tracking framework employing model selection to handle the complex data association problems that arise in crowded scenes. This allows our system to apply category-specific motion models for robust tracking and prediction.

In order to cope with the challenges of real-world operation, we additionally introduce numerous couplings and feedback paths between the different components of our system. Thus, we jointly estimate the ground surface and supporting object detections and let both steps benefit from

each other. The resulting detections are transferred into world coordinates with the help of visual odometry and are grouped into candidate trajectories by the tracker. Successful tracks are then again fed back to stabilize visual odometry and depth computation through their motion predictions. Finally, the results are combined in a dynamic occupancy map such as the one shown in Fig. 1(bottom right), which allows free space computation for a later navigation module.

The main contribution of this paper is to show that vision based sensing has progressed sufficiently for such a system to become realizable. Specifically, we focus on tracking-by-detection of pedestrians in busy inner-city scenes, as this is an especially difficult but very important application area of future robotic and automotive systems. Our focus on vision alone does not preclude the use of other sensors such as LIDAR or GPS/INS—in any practical robotic system those sensors have their well-deserved place, and their integration can be expected to further improve performance. However, the richness of visual input makes it possible to infer very detailed semantic information about the target scene, and the relatively low sensor weight and cost make vision attractive for many applications.

The paper is structured as follows: the upcoming section reviews previous work. Section III then gives an overview of the the different components of our vision system with a focus on pedestrian tracking, before Section IV discusses its application to the generation of dynamic occupancy maps. Implementation details are given in Section V. Finally, we present experimental results on challenging urban scenarios in Section VI, before the paper is concluded in Section VII.

II. Related Work

Obstacle avoidance is one of the central capabilities of any autonomous mobile system. Many systems are building up occupancy maps [7] for this purpose. An exhaustive review can be found in [28]. While such techniques are geared towards static obstacles, a main challenge is to accurately detect moving objects in the scene. Such objects can be extracted independent of their category by modeling the shape of the road surface and treating everything that does not fit that model as an object (*e.g.* in [19], [26], [33]). However, such simple approaches break down in crowded situations where not enough of the ground may be visible. More accurate detections can be obtained by applying category-specific models, either directly on the camera images [5], [16], [25], [31], on the 3D depth information [1] or both in combination [9], [12], [27].

Tracking detected objects over time presents additional challenges due to the complexity of data association in crowded scenes. Targets are typically followed using classic tracking approaches such as Extended Kalman Filters (EKF), where data assignment is optimized using Multi-Hypothesis Tracking (MHT) [4], [22] or Joint Probabilistic Data Association Filters (JPDAF) [11]. Several robust approaches have been proposed based on those components either operating on depth measurements [23], [24], [29] or as tracking-by-detection approaches from purely visual input [13], [17], [31], [32]. The approach employed in this paper is based on our previous work [17]. It works online and simultaneously optimizes detection and trajectory estimation for multiple interacting objects and over long time windows, by operating in a hypothesis selection framework.

Fig. 2. Mobile recording platforms used in our experiments. Note that in this paper we only employ image information from a stero camera pair and do not make use of other sensors such as GPS or LIDAR.

III. System

Our vision system is designed for a mobile platform equipped with a pair of forward-looking cameras. Altogether, we report experimental results for three different such platforms, shown in Fig. 2. In this paper, we only use visual appearance and stereo depth, and integrate different components for ground plane and ego-motion estimation, object detection, tracking, and occupied area prediction.

Fig. 3(a) gives an overview of the proposed vision system. For each frame, the blocks are executed as follows. First, a depth map is calculated and the new frame's camera pose is predicted. Then objects are detected together with the supporting ground surface, taking advantage of appearance, depth, and previous trajectories. The output of this stage, along with predictions from the tracker, helps stabilize visual odometry, which updates the pose estimate for the platform and the detections, before running the tracker on these updated detections. As a final step, we use the estimated trajectories in order to predict the future locations for dynamic objects and fuse this information with a static occupancy map. The whole system is held entirely causal, *i.e.* at any point in time it only uses information from the past and present.

For the basic tracking-by-detection components, we rely on the framework described in [8]. The main contribution of this paper is to extend this framework to the prediction of future spatial occupancy for both static and dynamic objects. The following subsections describe the main system components and give details about their robust implementation.

A. Coupled Object Detection and Ground Plane Estimation

Instead of directly using the output of an object detector for the tracking stage, we introduce scene knowledge to reduce false positives. For this, we assume a simple scene model where all objects of interest reside on a common ground plane. As a wrong estimate of this ground plane has far-reaching consequences for all later stages, we try to avoid making hard decisions here and instead model the coupling between object detections and the scene geometry

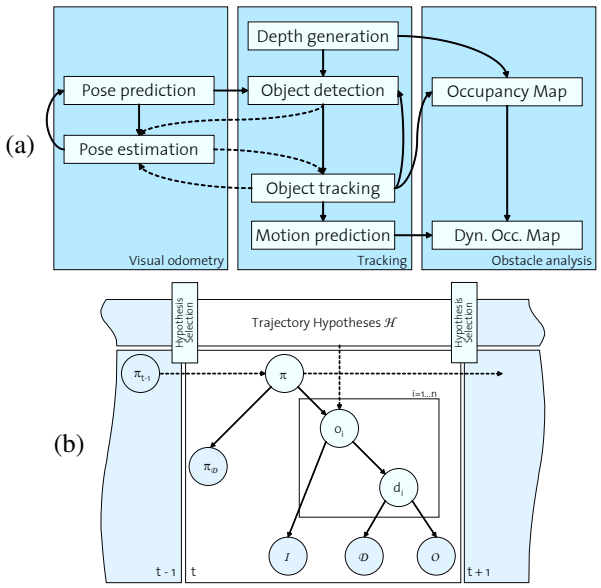

Fig. 3. (a) Flow diagram for our vision system. (b) Graphical model for tracking-by-detection with additional depth information (see text for details).

probabilistically using a Bayesian network (see Fig. 3(b)). This network is constructed for each frame and models the dependencies between object hypotheses o_i, object depth d_i, and the ground plane $\boldsymbol{\pi}$ using evidence from the image $\mathcal{I}$, the depth map $\mathcal{D}$, a stereo self-occlusion map $\mathcal{O}$, and the ground plane evidence $\boldsymbol{\pi}_{\mathcal{D}}$ in the depth map. Following standard graphical model notation, the plate indicates repetition of the contained parts for the number of objects n.

In this model, an object's probability depends both on its geometric world position and size (expressed by $P(o_i|\boldsymbol{\pi})$), on its correspondence with the depth map $P(o_i|d_i)$, and on $P(\mathcal{I}|o_i)$, the object likelihood estimated by the object detector. The likelihood $P(\boldsymbol{\pi}_{\mathcal{D}}|\boldsymbol{\pi})$ of each candidate ground plane is modeled by a robust estimator taking into account the uncertainty of the inlier depth points. The prior $P(\boldsymbol{\pi})$, as well as the conditional probability tables, are learned from a training set.

In addition, we introduce temporal dependencies, indicated by the dashed arrows in Fig. 3(b). For the ground plane, we propagate the state from the previous frame as a temporal prior $P(\boldsymbol{\pi}|\boldsymbol{\pi}_{t-1}) = (1-\alpha)P(\boldsymbol{\pi}) + \alpha P(\boldsymbol{\pi}_{t-1})$ that stabilizes the per-frame information from the depth map $P(\boldsymbol{\pi}_{\mathcal{D}}|\boldsymbol{\pi})$. For the detections, we add a spatial prior for object locations that are supported by tracked candidate trajectories $\mathcal{H}_{t_0:t-1}$. As shown in Fig. 3(b), this dependency is not a first-order Markov chain, but reaches many frames into the past, as a consequence of the tracking framework explained in Section III-B.

The advantage of this Bayesian network formulation is that it can operate in both directions. Given a largely empty scene where depth estimates are certain, the ground plane can significantly constrain object detection. In more crowded situations where less of the ground is visible, on the other hand, the object detector provides sufficient evidence to assist ground plane estimation.

B. Tracking, Prediction

After passing the Bayesian network, object detections are placed into a common world coordinate system using camera positions estimated from visual odometry. The actual tracking system follows a multi-hypotheses approach, similar to the one described in [17]. We do not rely on background modeling, but instead accumulate the detections of the current and past frames in a space-time volume. This volume is analyzed by growing many trajectory hypotheses using independent bi-directional Extended Kalman filters (EKFs) with a holonomic constant-velocity model. While the inclusion of further motion models, as *e.g.* done in [27], would be possible, it proved to be unnecessary in our case.

By starting EKFs from detections at different time steps, an overcomplete set of trajectories is obtained, which is then pruned to a minimal consistent explanation using model selection. This step simultaneously resolves conflicts from overlapping trajectory hypotheses by letting trajectories compete for detections and space-time volume. In a nutshell, the pruning step employs quadratic pseudo-boolean optimization to pick the set of trajectories with maximal joint probability, given the observed evidence over the past frames. This probability

- increases as the trajectories explain more detections and as they better fit the detections' 3D location and 2D appearance through the individual contribution of each detection;
- decreases when trajectories are (partially) based on the same object detections through pairwise corrections to the trajectories' joint likelihoods (these express the constraints that each pedestrian can only follow one trajectory and that two pedestrians cannot be at the same location at the same time);
- decreases with the number of required trajectories through a prior favoring explanations with fewer trajectories – balancing the complexity of the explanation against its goodness-of-fit in order to avoid over-fitting ("Occam's razor").

For the mathematical details, we refer to [17]. The most important features of this method are automatic track initialization (usually, after about 5 detections) and the ability to recover from temporary track loss and occlusion.

The selected trajectories $\mathcal{H}$ are then used to provide a spatial prior for object detection in the next frame. This prediction has to take place in the world coordinate system, so tracking critically depends on an accurate and smooth egomotion estimate.

C. Visual Odometry

To allow reasoning about object trajectories in the world coordinate system, the camera position for each frame is estimated using visual odometry. The employed approach builds upon previous work by [8], [20]. In short, each incoming image is divided into a grid of 10×10 bins, and an approximately uniform number of points is detected

Fig. 4. Visual odometry and occupancy maps are only based on image parts not explained by tracked objects, i.e. the parts we believe to be static. Left: original image with detected features. Right: image when features on moving objects (green) are ignored.

in each bin using a Harris corner detector with locally adaptive thresholds. The binning encourages a feature distribution suitable for stable localization. To reduce outliers in RANSAC, we mask out corners that coincide with predicted object locations from the tracker output and are hence not deemed suitable for localization, as shown in Fig. 4.

In the initial frame, stereo matching and triangulation provide a first estimate of the 3D structure. In subsequent frames, we use 3D-2D matching to get correspondences, followed by camera resection (3-point pose) with RANSAC. Old frames ($t' < t-15$) are discarded, along with points that are only supported by those removed frames. To guarantee robust performance, we introduce an explicit failure detection mechanism based on the covariance of the estimated camera position, as described in [8]. In case of failure, a Kalman filter estimate is used instead of the measurement, and the visual odometry is restarted from scratch. This allows us to keep the object tracker running without resetting it. While such a procedure may introduce a small drift, a locally smooth trajectory is more important for our application. In fact, driftless global localization would require additional input from other sensors such as a GPS.

IV. Occupancy Map and Free Space Prediction

For actual path planning, the construction of a reliable occupancy map is of utmost importance. We split this in two parts according to the static scene and the dynamically moving objects.

Static Obstacles. For static obstacles, we construct a stochastic occupancy map based on the algorithm from [2]. In short, incoming depth maps are projected onto a polar grid on the ground and are fused with the integrated and transformed map from the previous frames. Based on this, free space for driving can be computed using dynamic programming. While [2] integrate entire depth maps (including any dynamic objects) for the construction of the occupancy map, we opt to filter out these dynamic parts. As in the connection with visual odometry, we use the tracker prediction as well as the current frame's detections to mask out any non-static parts. The reasons for this are twofold: first, integrating non-static objects can result in a smeared occupancy map. Second, we are not only interested in the *current* position of the dynamic parts, but also in their *future* locations. For this, we can use accurate and category-specific motion models inferred from the tracker.

Dynamic Obstacles. As each object selected by the tracker is modeled by an independent EKF, we can predict its future position and obtain the corresponding uncertainty $\mathtt{C}$. Choosing a bound on the positional uncertainty then yields an ellipse where the object will reside with a given probability. In our experiments, a value of 99% resulted in a good compromise between safety from collision and the need to leave a navigable path for the robot to follow. For the actual occupancy map, we also have to take into consideration the object's dimensions and, in case of an anisotropic "footprint", the bounds for its rotation. We assume pedestrians to have a circular footprint, so the final occupancy cone can be constructed by adding the respective radius to the uncertainty ellipse. In our visualization, we show the entire occupancy cone for the next second, *i.e.* the volume the pedestrian is likely to occupy within that time.

Based on this predicted occupancy map, free space for driving can be computed with the same algorithm as in [2], but using an appropriate prediction horizon. Note that in case a person was not tracked successfully, it will still occur in the static occupancy map, as a sort of graceful degradation of the system.

V. Detailed Implementation

The system's parameters were trained on a sequence with 490 frames, containing 1'578 annotated pedestrian bounding boxes. In all experiments, we used data recorded at a resolution of 640×480 pixels (bayered) at 13–14 fps, with a camera baseline of 0.4 and 0.6 meters for the child stroller and car setups, respectively.

Ground Plane. For training, we infer the ground plane directly from $\mathcal{D}$ using Least-Median-of-Squares (LMedS), with bad estimates discarded manually. Related but less general methods include *e.g.* the *v*-disparity analysis [15]. For tractability, the ground plane parameters (θ, ϕ, π_4) are discretized into a $6\times6\times20$ grid, with bounds inferred from the training sequences. The training sequences also serve to construct the prior distribution $P(\boldsymbol{\pi})$.

Object Hypotheses. Our system is independent of a specific detector choice. In the experiments presented here, we use a publicly available detector based on a Histogram-of-Oriented-Gradients representation [5]. The detector is run with a low confidence threshold to retain the necessary flexibility—in the context of the additional evidence we are using, final decisions based only on appearance would be premature. The range of detected scales corresponds to pedestrian heights of 60–400 pixels. The object size distribution is modeled as a Gaussian $\mathcal{N}(1.7, 0.085^2)$ [m], as in [14]. The depth distribution is assumed uniform in the system's operating range of 0.5–30 [m], respectively 60 [m] for the car setup.

Depth Cues. The depth map $\mathcal{D}$ for each frame is obtained with a publicly available, belief-propagation-based disparity estimation software [10]. All results reported in this paper

are based on this algorithm. In the meantime, we have also experimented with a fast GPU-based depth estimator, which seems to achieve similar system-level accuracy. However, we still have to verify those results in practice. For verifying detections by depth measurements in the Bayesian network, we consider the agreement of the measured mean depth inside the detection bounding box with the ground-plane distance to the bounding box foot-point. As the detector's bounding box placement is not always accurate, we allow the Bayesian network to "wiggle around" the bounding boxes slightly in order to improve goodness of fit. The final classifier for an object's presence is based on the number of inlier depth points and is learned from training data using logistic regression.

Belief Propagation. The network of Fig. 3 is constructed for each frame, with all variables modeled as discrete entities and their conditional probability tables defined as described above. Inference is conducted using Pearl's Belief Propagation [21]. For efficiency reasons, the set of possible ground planes is pruned to the 20% most promising ones (according to prior and depth information).

VI. Results

In order to evaluate our vision system, we applied it to three test sequences, showing strolls and drives through busy pedestrian zones. The sequences were acquired with the platforms seen in Fig. 2.[1] The first test sequence ("Seq. #1"), recorded with platform (a), shows a walk over a crowded square, extending over 230 frames. The second sequence ("Seq. #2"), recorded with platform (b) at considerably worse image contrast, contains 5'193 pedestrian annotations in 999 frames. The third test sequence ("Seq. #3") consists of 800 frames and was recorded from a car passing through a crowded city center, where it had to stop a few times to let people pass. We annotated pedestrians in every fourth frame, resulting in 960 annotations for this sequence.

For a quantitative evaluation, we measure bounding box overlap in each frame and plot recall over false positives per image for three stages of our system. The results of this experiment are shown in Fig. 5(left, middle). The plot compares the raw detector output, the intermediate output of the Bayesian network, and the final tracking output. As can be seen, discarding detections that are not in accordance with the scene by the Bayesian network greatly reduces false positives with hardly any impact on recall. The tracking stage additionally improves the results and in most cases achieves a higher performance than the raw detector. It should be noted, though, that a single-frame comparison is not entirely fair here, since the tracker requires some detections to initialize (losing recall) and reports tracking results through occlusions (losing precision if the occluded persons are not annotated). However, the tracking stage provides the necessary temporal information that makes the entire motion prediction system at all possible. The blue curves in Fig. 5 show the performance on all annotated pedestrians. When only considering the immediate range up to 15m distance (which is suitable for a speed of 30 km/h in inner-city scenarios), performance is considerably better, as indicated by the red curves.

To assess the suitability of our system for path planning, we investigate the precision of the motion prediction for increasing time horizons. This precision is especially interesting, since it allows to quantify the possible advantage over system modeling only static obstacles. Specifically, we compare the bounding boxes obtained from the tracker's prediction with the actual annotations in the frame and count the fraction of false positives ($1 - \mathrm{prec}$). The results can be seen in Fig. 5(right). As expected, precision drops with increasing lookahead time, but stays within acceptable limits for a prediction horizon $\leq$ 1s (12 frames). Note that this plot should only be taken qualitatively: a precision of 0.9 does not imply an erroneous replanning every 10th frame, as many of the predicted locations do not affect the planned path. Rather, this experiment shows that for reasonable prediction horizons, the precision does not drop considerably.

Example tracking results for Seq. #1 are shown in Fig. 6. The operating point for generating those results was the same as the one used in Fig. 5(right). Recorded on a busy city square, many people interact in this scene, moving in all directions, stopping abruptly (*e.g.* the first orange box), and frequently occluding each other (see *e.g.* the second orange box). The bounding boxes are color coded to show the tracked identities (due to the limited palette, some color labels repeat). Below each image, we show the inferred dynamic obstacle map in an overhead view. Static obstacles are marked in black; each tracked pedestrian is entered with its current position and the predicted occupancy cone for the next second (for standing pedestrians, this cone reduces to a circle). As can be seen, our system is able to track most of the visible pedestrians correctly and to accurately predict their future motion.

Fig. 7 shows more results for Seq. #2. Note that both adults and children are identified and tracked correctly even though they differ considerably in their appearance. In the bottom row of the figure, a man in pink walks diagonally towards the camera. Without motion prediction, a following navigation module might issue an unnecessary stop here. However, our system correctly determines that he presents no danger of collision and resolves this situation. Also note how the standing woman in the white coat gets integrated into the static occupancy map as soon as she is too large to be detected. This is a safe fallback in the design of our system—when no detections are available, its results simply revert to those of a depth-integration based occupancy map.

Finally, Fig. 8 demonstrates the vision system in a car application. Compared to the previous sequences, the viewpoint is quite different, and faster scene changes result in fewer data points for creating trajectories. Still, stable tracking performance can be obtained also for quite distant pedestrians.

System Performance Apart from the object detectors, the

[1]Data and videos are available on `http://www.vision.ee.ethz.ch/~aess/icra2009/`.

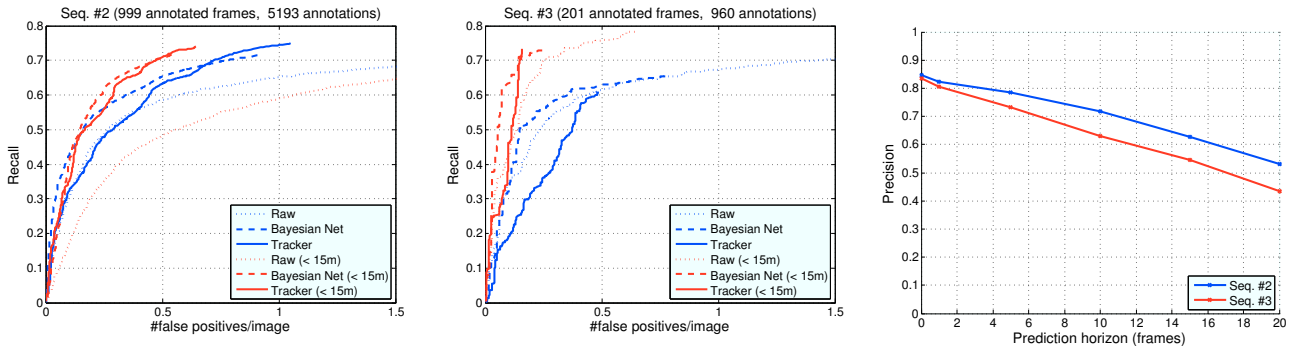

Fig. 5. **Left, middle:** Performance plots of our system on two test sequences. We plot the overall recognition performance, as well as the obtained performance within a range of 15m, the range that we consider important for autonomous driving at low speeds (30 km/h). **Right:** Precision of the tracker prediction for increasing prediction horizon. Data was recorded at 12–14 fps.

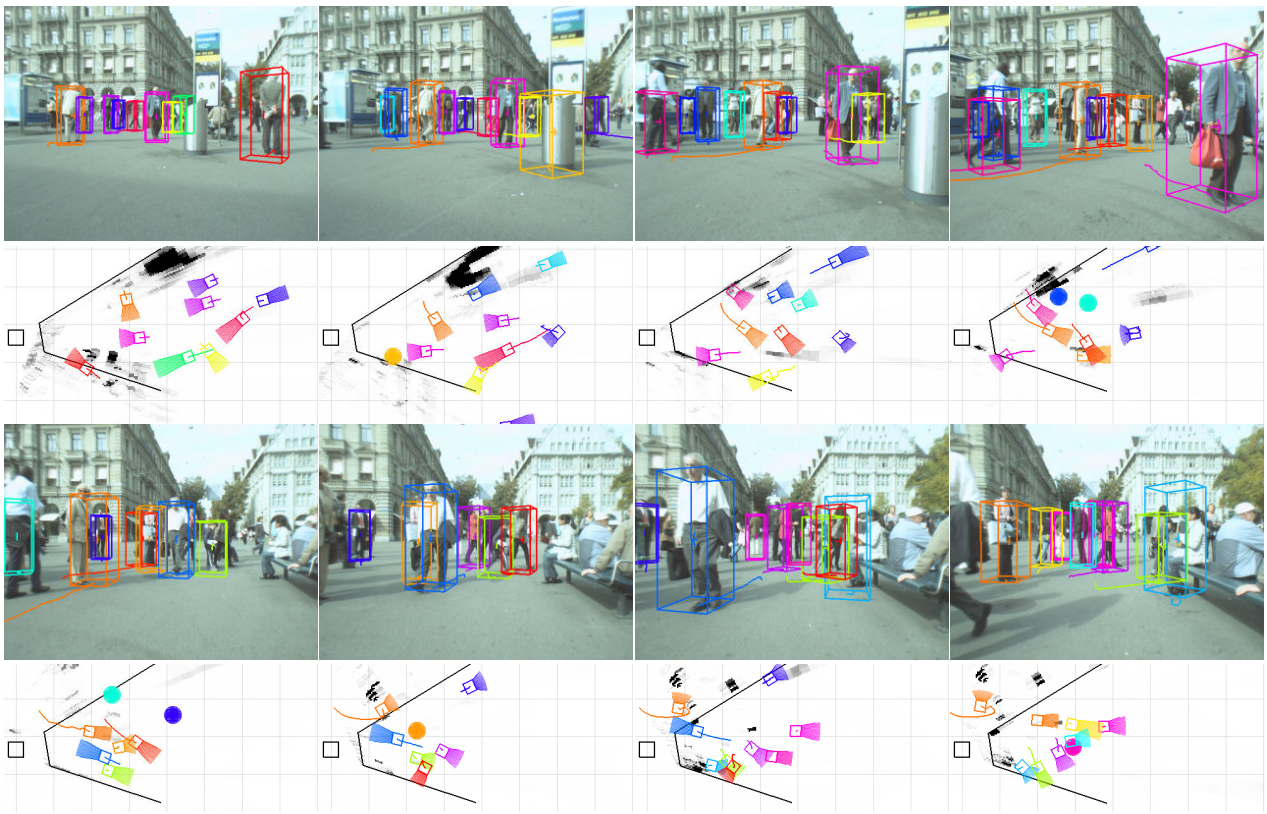

Fig. 6. Example tracking results for Seq. #1. For each image, we show the actual tracking results as well as an overhead view of the dynamic occupancy map.

entire system is implemented in an integrated fashion in C/C++, with several procedures taking advantage of GPU processing. For the complex parts of Seq. #3 (15 simultaneous objects), we can achieve processing times of around 400 ms per frame on an Intel Core2 CPU 6700, 2.66GHz, nVidia GeForce 8800 (see Tab. I). While the detector stage is the current bottleneck (the detector was run offline and needed about 30 seconds per image), we want to point out that for the HOG detector, real-time GPU implementations exist [30], which could be substituted to remove this restriction.

Component	GPU	CPU	Time
Detector		×	2 × 30 s
Depth map (old)		×	2 × 20 s
Depth map (new)	×		2 × 20 ms
Bayesian network		×	150 ms
Visual odometry	×	×	40 ms
Tracker		×	150 ms

TABLE I

PROCESSING TIMES OF THE VARIOUS COMPONENTS IN OUR SYSTEM.

VII. CONCLUSION

In this paper, we have presented a mobile vision system for the creation of dynamic obstacle maps for automotive or mo-

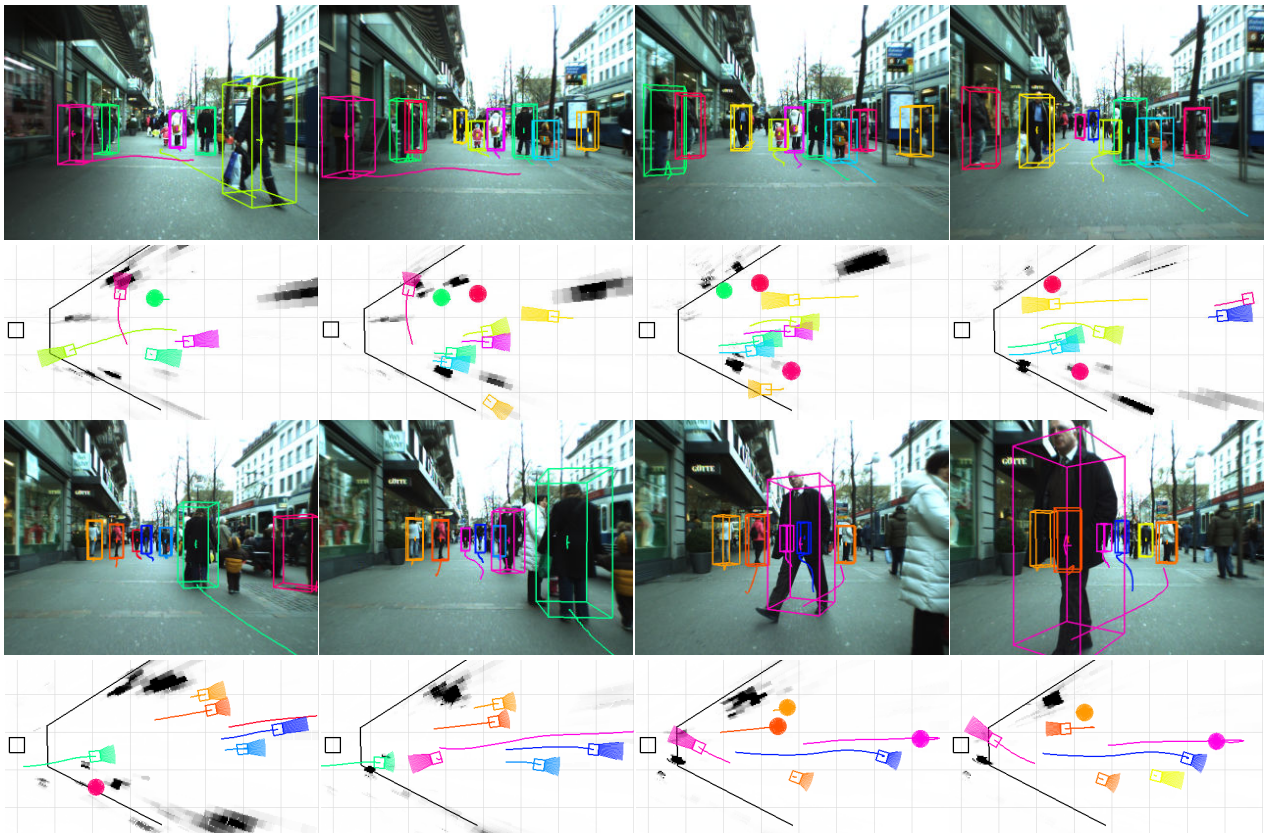

Fig. 7. Example tracking results for Seq. #2. Note the long trajectories and the tracker's ability to handle temporary occlusions in complex scenarios.

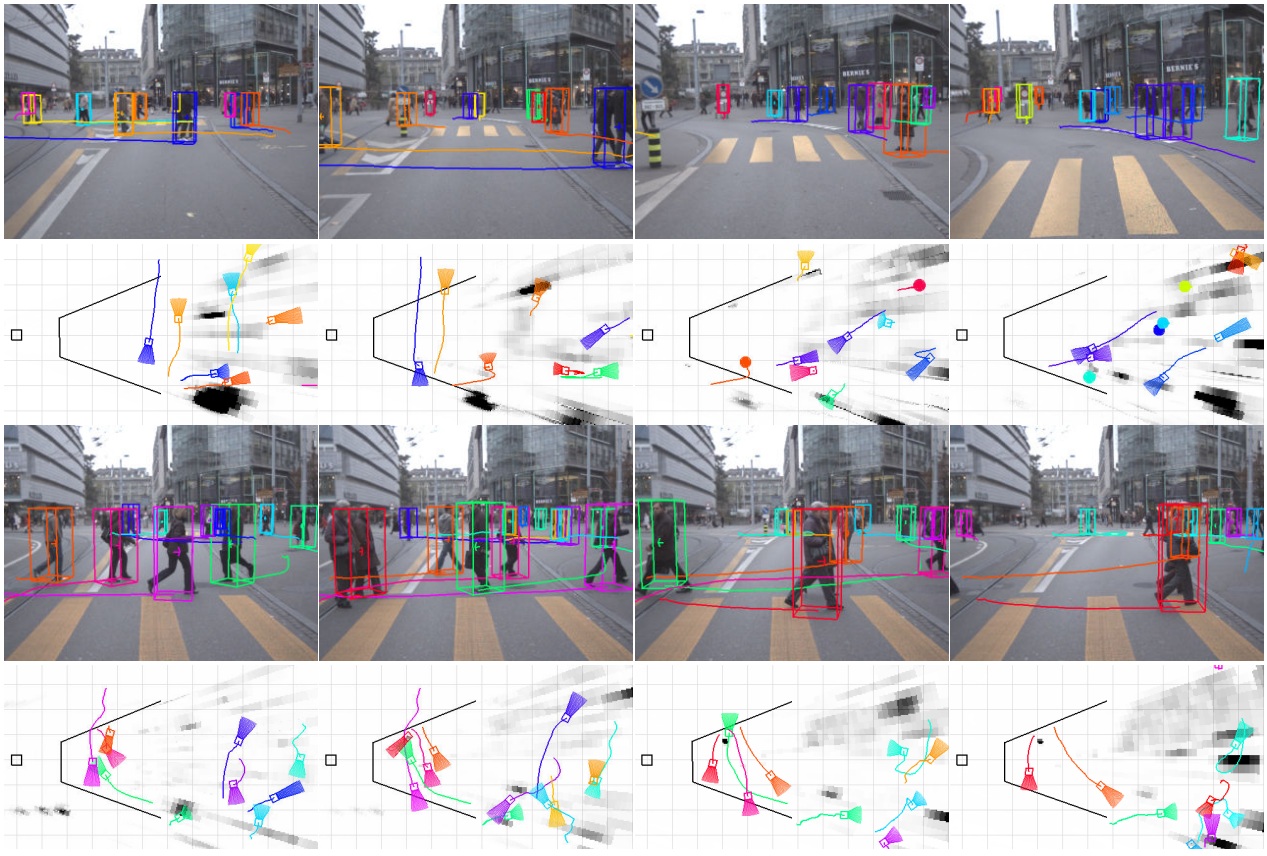

Fig. 8. Example tracking results for the third test sequence, recorded from a car.

bile robotics platforms. Such maps should provide valuable input for actual path planning algorithms [18]. Our approach relies on a robust tracking system that closely integrates different modules (appearance-based object detection, depth estimation, tracking, and visual odometry). To resolve the complex interactions that occur between pedestrians in urban scenarios, a multi-hypothesis tracking approach is employed. The inferred predictions can then be used to extend a static occupancy map generation system to a dynamic one, which then allows for more detailed path planning. The resulting system can handle very challenging scenes and delivers accurate predictions for many simultaneously tracked objects.

In future work, we plan to optimize the individual system components further with respect to run-time and performance. As discussed before, system operation at 2-3 fps is already reachable now, but additional improvements are necessary for true real-time performance. In addition, we plan to improve the trajectory analysis by including more elaborate motion models and to combine it with other sensing modalities such as GPS and LIDAR.

Acknowledgments. This project has been funded in parts by Toyota Motor Corporation/Toyota Motor Europe and the EU projects DIRAC (IST-027787) and EUROPA (ICT-2008-231888).

REFERENCES

[1] K. O. Arras, O. M. Mozos, and W. Burgard. Using boosted features for the detection of people in 2d range data. In *ICRA*, 2007.

[2] H. Badino, U. Franke, and R. Mester. Free space computation using stochastic occupancy grids and dynamic programming. In *ICCV Workshop on Dynamical Vision (WDV)*, 2007.

[3] M. Betke, E. Haritaoglu, and L. S. Davis. Real-time multiple vehicle tracking from a moving vehicle. *MVA*, 12(2):69–83, 2000.

[4] I. J. Cox. A review of statistical data association techniques for motion correspondence. *IJCV*, 10(1):53–66, 1993.

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

[6] DARPA. DARPA urban challenge rulebook. In *Webpage*, 2008. `http://www.darpa.mil/GRANDCHALLENGE/docs/Urban_Challenge_Rules_102707.pdf`.

[7] A. Elfes. Sonar-based real-world mapping and navigation. *IEEE Journal of Robotics and Automation*, 3(3):249–265, 1987.

[8] A. Ess, B. Leibe, K. Schindler, and L. van Gool. A mobile vision system for robust multi-person tracking. In *CVPR*, 2008.

[9] A. Ess, B. Leibe, and L. van Gool. Depth and appearance for mobile scene analysis. In *ICCV*, 2007.

[10] P. F. Felzenszwalb and D. P. Huttenlocher. Efficient belief propagation for early vision. *IJCV*, 70:41–54, 2006.

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

[12] D. Gavrila and V. Philomin. Real-time object detection for "smart" vehicles. In *ICCV*, pages 87–93, 1999.

[13] D. M. Gavrila and S. Munder. Multi-cue pedestrian detection and tracking from a moving vehicle. *IJCV*, 73:41–59, 2007.

[14] D. Hoiem, A. A. Efros, and M. Hebert. Putting objects in perspective. In *CVPR*, 2006.

[15] R. Labayrade, D. Aubert, and J.-P. Tarel. Real time obstacle detection on non flat road geometry through 'v-disparity' representation. In *IVS*, 2002.

[16] B. Leibe, A. Leonardis, and B. Schiele. Robust object detection with interleaved categorization and segmentation. *International Journal of Computer Vision*, 77(1-3):259–289, May 2008.

[17] B. Leibe, K. Schindler, N. Cornelis, and L. Van Gool. Coupled detection and tracking from static cameras and moving vehicles. *IEEE TPAMI*, 30(10):1683–1698, 2008.

[18] K. Macek, A. D. Vasquez, T. Fraichard, and R. Siegwart. Safe vehicle navigation in dynamic urban scenarios. In *ITSC*, 2008.

[19] S. Nedevschi, R. Danescu, D. Frentiu, T. Graf, and R. Schmidt. High accuracy stereovision approach for obstacle detection on non-planar roads. In *Proc IEEE Intelligent Engineering Systems*, 2004.

[20] D. Nistér, O. Naroditsky, and J. R. Bergen. Visual odometry. In *CVPR*, 2004.

[21] J. Pearl. *Probabilistic Reasoning in Intelligen Systems*. Morgan Kaufmann Publishers Inc., 1988.

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

[23] M. Scheutz, J. McRaven, and G. Cserey. Fast, reliable, adaptive, bimodal people tracking for indoor environments. In *IROS*, 2004.

[24] D. Schulz, W. Burgard, D. Fox, and A. Cremers. People tracking with mobile robots using sample-based joint probabilistic data association filters. *IJRR*, 22(2):99–116, 2003.

[25] A. Shashua, Y. Gdalyahu, and G. Hayun. Pedestrian detection for driving assistance systems: Single-frame classification and system level performance. In *IVS*, 2004.

[26] M. Soga, T. Kato, M. Ohta, and Y. Ninomiya. Pedestrian detection with stereo vision. In *IEEE International Conf. on Data Engineering*, 2005.

[27] L. Spinello, R. Triebel, and R. Siegwart. Multimodal people detection and tracking in crowded scenes. In *Proc. of The AAAI Conference on Artificial Intelligence (Physically Grounded AI Track)*, July 2008.

[28] S. Thrun. *Probabilistic Robotics*. The MIT Press, 2005.

[29] C.-C. Wang, C. Thorpe, and S. Thrun. Online simultaneous localization and mapping with detection and tracking of moving objects: Theory and results from a ground vehicle in crowded urban areas. In *ICRA*, 2003.

[30] C. Wojek, G. Dorkó, A. Schulz, and B. Schiele. Sliding-windows for rapid object class localization: A parallel technique. In *DAGM*, 2008.

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

[32] L. Zhang, Y. Li, and R. Nevatia. Global data association for multi-object tracking using network flows. In *CVPR*, 2008.

[33] L. Zhao and C. Thorpe. Stereo- and neural network-based pedestrian detection. In *ITS*, 2000.