Real-Time Multi-Person Tracking with Time-Constrained Detection

Dennis Mitzel
mitzel@umic.rwth-aachen.de
Patrick Sudowe
sudowe@umic.rwth-aachen.de
Bastian Leibe
leibe@umic.rwth-aachen.de

UMIC Research Centre
RWTH Aachen University
Aachen, GERMANY

Abstract

This paper presents a robust real-time multi-person tracking framework for busy street scenes. Tracking-by-detection approaches have recently been successfully applied to this task. However, their run-time is still limited by the computationally expensive object detection component. In this paper, we therefore consider the problem of making best use of an object detector with a fixed and very small time budget. The question we ask is: given a fixed time budget that allows for detector-based verification of k small regions-of-interest (ROIs) in the image, what are the best regions to attend to in order to obtain stable tracking performance? We address this problem by applying a statistical Poisson process model in order to rate the urgency by which individual ROIs should be attended to. These ROIs are initially extracted from a 3D depth-based occupancy map of the scene and are then tracked over time. This allows us to balance the system resources in order to satisfy the twin goals of detecting newly appearing objects, while maintaining the quality of existing object trajectories.

1 Introduction

In this paper we address the problem of vision-based multi-person tracking in busy urban environments using a camera setup mounted on a moving vehicle, *e.g.* an autonomous mobile robot. Recent years have seen considerable progress in this area, fueled by the development of advanced tracking-by-detection approaches [1, 8, 10, 14, 26]. However, those approaches require a robust object detector, which is triggered for each frame to detect all target objects in the scene. Although efficient CPU-based [19] and GPU-based [17, 25] detectors have been proposed for this purpose, their requirements with respect to computational power and energy consumption are not yet satisfactory for use on autonomous platforms.

Approaches targeted at automotive scenarios have had to deal with this problem for a long time. They usually restrict detector evaluation to a small number of pre-selected ROIs [11] based on 3D geometry [3], motion [5], texture content [21], or stereo depth [10]. Recent approaches targeted at mobile robotics have adopted similar strategies [1, 2]. However, such approaches risk losing detections if the corresponding regions are missed by the ROI selection stage. What makes matters worse, the question which ROIs to select is usually addressed independently for every frame [11]. This results in a suboptimal selection strategy,

either risking to lose important detections, or spreading the detector's time budget over many regions that have already been verified as containing or not containing an object before.

In contrast, we consider the case of an object detector with a fixed time budget *in the context of a tracking system*. We also assume that the detector can only process a small number of ROIs in each frame, but we balance the ROI selection over time, such that at each time instant, only those ROI candidates are considered for which attention is most urgently required in order to produce stable tracking results. The question we pose is: given a detector with a budget to attend to k ROIs in each frame and a cheap low-level tracking system to follow ROI candidates over time, which ones should be selected? To address this question, we propose the following approach. We first create ROI candidates from a depth map of the scene and from already existing object trajectories. These candidates are associated and tracked over time using local depth and appearance information. We then model the selection process of k ROIs to be verified by the detector using a statistical Poisson process model. Briefly stated, this model associates each tracked ROI candidate with a low probability of causing an important event. For regions in the background, this event means that the region now contains a person, despite of this having previously been verified as not being the case. For regions on tracked person trajectories, the event indicates a tracking failure that causes the low-level tracker to drift. In both cases, the occurrence of an event has the consequence that the region should be attended to and be verified by the object detector. Since we cannot predict where those events will happen, we model their probability of occurrence using a Poisson process. The result of this process indicates the *urgency* by which the detector should attend to a region in order to limit the probability of the event influencing the tracking results. In our approach, the urgency of a region is additionally moderated by its *utility* for maintaining tracking performance, which gives preference to regions close to the camera.

In order to separate the different effects of foreground and background regions (*i.e.*, regions stemming from already existing trajectories and regions in which no person has been found yet), we propose to apply a two-tiered model. For foreground regions, the Poisson process accumulates the uncertainty of the individual tracking steps, while it assumes a fixed event occurrence rate for the background regions. In addition, we extend the model with a special treatment for trajectories that are predicted to emerge from an occlusion. As such an event requires immediate attention in order to apply potential corrections, we always give preference to such regions. Once the selected ROIs have been verified by the detector, its output is converted to 3D world coordinates using the camera position from Structure-from-Motion (SfM), together with an estimate of the ground plane. We then integrate the 3D measurements in a multi-hypothesis tracking approach similar to [14]. As our experimental results will demonstrate, our approach reaches state-of-the-art performance with high tracking quality, even with a significantly reduced time budget for the detector. We experimentally investigate the time budget required for robust system-level performance and show that employing the stochastic Poisson process model optimizes ROI selection, such that only three detector evaluations per frame are sufficient for obtaining a highly robust tracking system.

In summary, our paper makes the following contributions: (1) We demonstrate how ROI selection can be optimized in general by employing a Poisson process model and how this model can be adapted for a tracking-by-detection approach. (2) In order to satisfy the conflicting goals of detecting new objects while stabilizing already existing tracks, we propose a two-tiered realization of the Poisson process model that takes into account a track's accumulated uncertainty. (3) We experimentally show that the proposed framework achieves robust multi-person tracking performance even with few ROI detector evaluations, making it possible to reduce detector evaluation to a minimum.

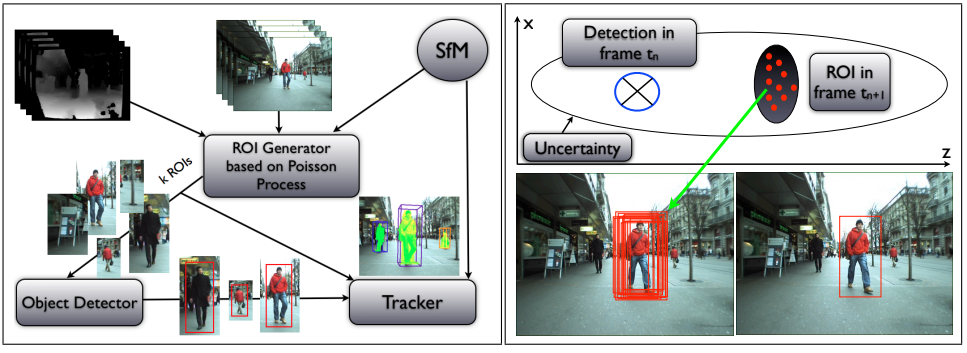


Figure 1: (left) Overview over the different components of our tracking system and their connections. (right) Visualization of how we create new observations for the EKF based on the detection from the last frame and the extracted ROI candidates. For each detection from the last frame, we sample points from ROI candidates which are inside a certain uncertainty region around the last detection. The final observation is computed by the mean of sampled points which are weighted corresponding to the Bhattacharyya distance between the last detection and the back-projection of the individual sampled position.

2 Related Work

Multi-object tracking is an important capability for vision applications in mobile robotics and autonomous vehicles [10]. The development of powerful object detectors [4] has made robust multi-person tracking-by-detection approaches feasible in challenging inner-city scenarios [1, 7, 10, 14, 26]. A disadvantage of pure tracking-by-detection approaches is however the requirement of running a computationally expensive object detector for each frame and sliding it over the entire image, even though only a small fraction of the considered image locations actually contain persons.

Many object detection approaches targeted at real-time applications follow a simple strategy of extracting ROIs based on motion [5], texture content [20] and stereo depth [1, 2, 11, 13] in order to reduce the detector evaluation time. In our approach we follow the strategy of extracting the ROI candidates from stereo depth data similar to [1]. In contrast to [1], we however do not run the detector for all given ROI candidates per frame, since inner-city scenes contain many unwanted objects (*e.g.* trees, buildings, signs, trash bins) that are also potential ROI candidates. As the results from [1] show, their strategy is more applicable to open land scenes with few potential ROI candidates.

There are also hybrid approaches that try to reduce the number of frames for which the detector needs to be evaluated by applying a low level image based tracker [16] that tracks detected persons based just on foreground and background appearance models. We follow a similar strategy in propagating candidate ROIs and tracked persons using low-level depth and appearance cues. However, the approach by [16] only reduces the frequency of detector evaluation on average – in busy scenes where low-level trackers degrade fast, it may still need to run the detector in every frame. As a result, this approach is less suitable for hard real-time systems, where a fixed time budget needs to be kept.

The task of selecting ROIs is closely related to visual attention. Saliency based visual attention systems [9, 17] typically extract image areas which differ from surrounding distractors by their unique color and intensity. However, those approaches are not applicable for our task, where not only one individual object sticks out of the background, but where many people with potentially similar appearance need to be detected and reliably tracked.

Poisson process models were already applied for active scene exploration with pan-tilt-zoom cameras [21, 22, 23]. Those approaches model the chance of appearance of pedestrians in a certain area in order to decide whether to zoom inside this area and risk missing some occurrence of newly appearing persons in other areas. Our scenario is however different in that the Poisson process models are attached to moving objects, which periodically need to be re-attended, rather than to fixed areas.

3 System Overview

Fig. 1 (left) shows an overview of our proposed tracking system. The system consists of three major components: ROI candidate generation, object detection, and tracking, which will be explained in more detail in the following.

For extracting the ROI candidates, we rely on depth information, which we assume to be available nowadays in real-time through dedicated sensors (*e.g.*, Microsoft's Kinect) or hardware processing solutions (*e.g.*, [18]). In addition, we use visual odometry to estimate the camera vehicle's egomotion, and we estimate the scene ground plane in each frame. For both tasks, there are also real-time approaches available [13, 15]. For the purpose of this paper, we use the data generously provided with the datasets of [7].

Given a color image and a corresponding depth map, we extract ROI candidates as local maxima of the depth map points within a height corridor of two meters, projected onto the ground plane. For each new ROI candidate, a Poisson process is initialized modeling the urgency of verification of the ROI by the detector. The candidate regions of past frames are associated with the newly extracted ROI candidates in the current frame and the urgency is propagated to the new regions. Depending on the time budget defined for the detector, a certain number k of ROIs with the highest urgency is verified by the detector. The detector output then enables the multi-hypothesis tracker to initialize new trajectories or to extend the existing ones based on an Extended Kalman filter (EKF). Overall, this results in a robust tracking system running at more than 15 Hz and allowing to define a time budget for the computationally expensive object detector. The time budget is represented by the number k of ROIs which are verified in each frame.

4 Poisson Process Attention Model

A Poisson process is a stochastic process in which events occur continuously and independently of each other. Mathematically, the process is described by a collection of random variables $\{N(t) : t \geq 0\}$ where $N(t)$ is the number of events that have occurred up to time t. Given a rate parameter λ, if the interval times are independent and obey exponential distributions (Poisson distributions) $Exp(\lambda)$, then a Poisson process is formally defined as

$$P\{\text{interval time} > t\} = \exp(-\lambda t)$$

From this, one can directly derive the chance of an event to occur. This chance increases with each time step since the last event occurrence at t_0 and can be defined as:

$$p(T < (t - t_0)) = 1 - \exp(-\lambda (t - t_0)) \tag{1}$$

where T is the waiting time until the next event. As described above, we fix the time budget for the detector, such that it is only allowed to evaluate a certain number of ROI candidates. In order to resolve the question in which order the ROI candidates should be verified by the detector, we will in the following model the urgency for verifying the region of interest using a Poisson process.

Having a busy street scene scenario, where pedestrians show up regularly, we consider the occurrence of a person within an ROI as a random event. We model the waiting time T

until the next occurrence within ROI $\mathbf{r}$ by an exponential distribution with the occurrence rate λ. The chance of an occurrence having taken place always increases with each frame where the ROI $\mathbf{r}$ is not verified by the detector and is computed by Eq. 1. As we are interested in tracking all persons in the scene, this chance corresponds to the urgency by which the ROI should be attended to in order to detect newly appearing persons.

After generating new ROI candidates, we associate the ROIs from the last frame with the new ones and propagate the time when the previous ROI was evaluated to the new ROI, thus increasing the urgency for verifying this part of the image. More details on the association procedure for ROIs can be found in Section 5. For the remaining new ROIs that could not be associated to ROIs from the previous frame, we start a new Poisson process. The urgency for the new ROIs is set to the lowest value by setting $t_0 = t$. Consequently, the detector always attends the ROI candidates with the highest urgency.

So far, the described ROI association process is a so-called background process, where we try to find newly appearing persons in the background and start new tracks for them. In addition, we run a second Poisson process (foreground process) for each already existing trajectory, which incorporates the track consistency based on the appearance model. This step assures that already found trajectories do not get lost due to low fixed-time budgets for the detector. To this end, we run a non-homogeneous Poisson process, whose rate function can change over time, representing the appearance model's consistency of the track:

$$\lambda(t) = w_{\text{tr}} \sum_{i=t_0}^{t} (1 - bhatta(t_i)), \tag{2}$$

where w_{tr} is a weighting factor, t_0 the time since the last detector verification, and *bhatta* is the Bhattacharyya coefficient between the new region's color histogram and the last associated detection. When a region was evaluated by the detector, the urgency is reset to zero as $t_0 = t$ in Eq.1.

In addition to the Poisson process, we introduce a further *utility* factor, which weights the ROI candidates with respect to their distance to the camera as follows:

$$utility(\mathbf{r}) = 1 - \exp(-w_d / d_{cam})), \tag{3}$$

where w_d is a weighting parameter and d_{cam} represents the distance to the camera. The *utility* factor is necessary in order to give the detector a preference for attending close-by regions, which are important for tasks like collision avoidance or pedestrian safety.

In order to select the k ROIs for verification, both measures, *urgency* and *utility*, are combined for ROI $\mathbf{r}$ as:

$$w(\mathbf{r}) = 1 - \exp(-\lambda(t - t_l) - w_d / d_{cam})) \tag{4}$$

where t_l represents the frame where the ROI was last evaluated. The detector is then triggered only for the k ROIs with the highest weight $w(\mathbf{r})$.

5 System Realization

Depth based ROI Generation. The idea behind the ROI extraction using stereo data is to fix the attention of the detector only on the few regions which may contain a wanted object. This allows us to run the computationally expensive detector only on small image regions regarding only few scales, rather than sliding over the whole image and all possible scales, which is computationally very expensive.

The results of ROI generation are shown in Fig. 2 (left). Given a depth map and the ground plane, the 3D points are projected onto a 2D grid map, omitting the points which are more than 2 meters above the ground plane. This restriction on the height helps us to

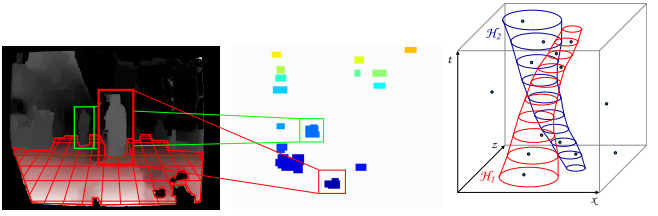


Figure 2: (left) An example of the stereo based ROI generation. The left image is the depth map. The middle image is the stereo range data projected onto the ground plane. (right) An example of EKF based trajectory generation. The blue hypothesis was started in the new frame and propagated backwards through previous frames. The red one is a hypothesis which was extended in the current frame.

reject points from overhanging parts of the scene, *e.g.* from buildings. Next, the grid cells are weighted with the distance to the camera, smoothed with an average filter, and thresholded by θ_{roi} in order to remove small noisy regions. The weighting is necessary, since farther objects consist of fewer points and thus would be removed by the thresholding without weighting them higher. Finally, we find the connected components on the grid map, which become the final ROI candidates. Each ROI candidate is represented by its center of mass and its width. Note that these ROIs are regions in 3D world coordinates. In order to obtain the corresponding image region, we take a rectangle with width of the ROI, height of 2 meters centered on the ROI's center of mass and parallel to the camera and project it into the image.

Object Detection. For pedestrian detection, we employ the popular HOG detector [4] in an efficient GPU implementation. Without any further constraints, our implementation [24] processes 640×480 images at 22Hz and 1280×960 images at 5Hz, while achieving the same detection performance as the original HOG detector [4]. When applied to full images, the GPU's power consumption however presents a serious limitation to its use for autonomous systems, restricting the vision system's battery life.

The advantage of our approach is that only small regions of interest need to be evaluated, rather than sliding over the entire image for all 27 scales the detector would consider for a 640×480 image resolution. Thus, in each frame we call the detector for evaluating k ROIs with 5 scales per ROI. The scales are determined as follows. The base scale is the result of dividing of the height of the ROI in image coordinates by 128 (the height of the sliding window of the HOG detector). To ensure the detection of pedestrians inside the ROI, we also consider two scales above the base scale and two scales below that are computed in multiplicative scale steps of 1.05. To summarize, we run the detector on small ROIs, which means computing the features only for those small regions and considering only five instead of 27 scales per region. This gives us an enormous speed-up – on average, only 2 ms computation time are required per ROI, instead of 44 ms if we slide over the entire image.

Tracking Model. The tracking model is an extended version of the multi-hypothesis tracking-by-detection system by [14]. In brief, the approach works as follows. The detector output is accumulated in a world coordinate system on the ground plane using the camera information estimated from SfM. The detections are linked to generate an over-complete set of competing trajectory hypotheses. For obtaining a subset of trajectories that best explains the collected measurements, we apply model selection in each frame.

Trajectory Generation. Linking the detections on the ground plane is done using an EKF with a constant-velocity model (Fig. 2 (right)). In each frame when new detections become available, we first try to extend already existing trajectories. In addition, new trajectory

hypotheses are generated by starting from the new detections and trying to grow them by applying the EKF backwards in time through the past detections in previous frames. Here we keep a trajectory's history for up to 100 frames. Due to the fact that a new detection is used for extension and also for generating new trajectories, each detection may end up in several competing trajectory hypotheses.

Hypothesis Selection. As a result of trajectory generation, we obtain an over-complete set of trajectory hypotheses. The score of each hypothesis is the combination of the likelihood of the assigned detections under the trajectory's motion and appearance model (represented by an RGB color histogram). The set of candidate trajectories is then pruned to a minimal consistent explanation using model selection in a Minimum Description Length framework, as presented in [14]. This step tries to resolve the conflicts between overlapping trajectories. For details of the mathematical formulation we refer to [14].

ROI-based Tracking. In contrast to [14], we set a fixed time budget for the detector by verifying only k small regions of interest. Hence, some of the measurements will be missed that are required for extending the existing trajectories. To cope with this problem, we additionally use the ROI candidates for generating measurements (observations) required for the EKF updating step. This is done in the following way (see Fig. 1 (right)). For frame t_{n+1}, we sample for each detection in frame t_n a number of $M = 20$ points randomly from the regions of interest in frame t_{n+1} that are within a certain uncertainty region around the detection. These points are back-projected to the image, generating possible detection bounding boxes. Next, we compute the appearance similarity based on RGB color histograms of the generated bounding boxes with the detection in frame t_n, employing the Bhattacharyya distance. The final detection is the mean of all sampled points weighted by the appearance similarity.

In some cases, due to noisy depth information, the new sampled detection is not correctly aligned to the pedestrian, causing the tracker to drift. By using the non-homogeneous Poisson process for each existing track, as described in Section 4, the drifting is detected through the appearance change. Thus, the urgency for a detector verification increases rapidly, resulting in the detector to be triggered for this area and the track to be revised. This is a crucial step in our tracking system, since we require at least three successive measurements for starting a trajectory, but once a ROI is evaluated, the urgency for that ROI is set to zero and as a consequence it will not be verified by the detector in the next frames.

ROI Propagation. In each frame, the newly extracted ROIs need to be associated with the ROIs from the last frame. To this end, we define a gating covariance that depends on the maximum velocity of a pedestrian. We assume that a pedestrian moves with at most $1.38 m/s$. Then given a new ROI and the covariance matrix $\Sigma = (0.4^2/(\text{fps}),\ 0; 0,\ 1.38^2/(\text{fps}))$, we associate only ROIs within the 0.95 confidence region. In case more than one ROI is inside the uncertainty region, we associate the new ROI with the closest one.

Occlusion Handling. For correct association of a person reappearing after a person-to-person occlusion, it is helpful to detect imminent occlusions first. To this end, we project the 3D prediction of the EKF of each tracked person into the image, computing the bounding box overlap. For persons with an overlap above 0.5, the occlusion is likely to occur and the person farther from the camera is marked as occluded. For the next 15 frames, we check whether the person is likely to reappear by performing the same bounding box check on its extrapolated EKF prediction. When a reappearance is likely, we create a virtual ROI at the predicted location with the urgency $u_{\max}$, forcing the detector to evaluate this region. This virtual ROI generation is necessary, since it is likely that no valid depth data will be available for the reappearing person due to stereo shadowing from the previously occluding person.

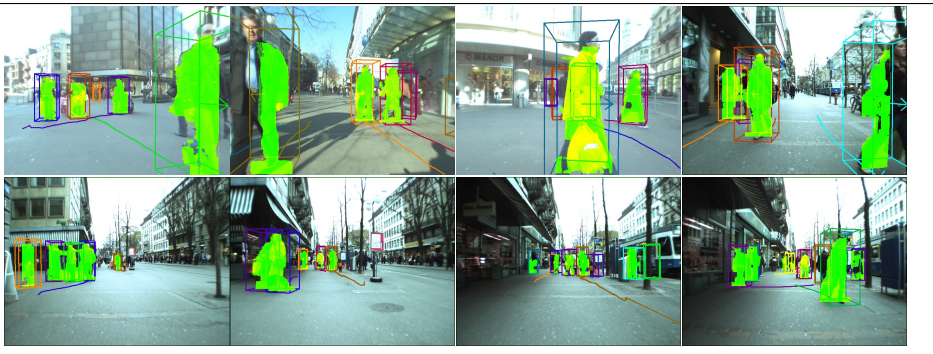


Figure 3: (first row) Capability to continue tracking close to the camera and/or the image borders. (second row) Example results on the test sequence BAHNHOF.

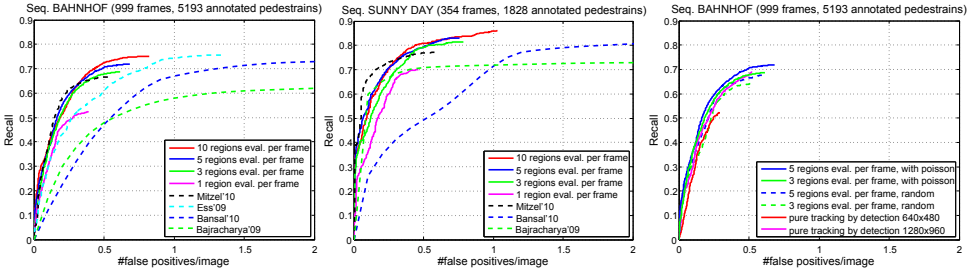
6 Experimental Results

In order to evaluate our approach, we applied it to two challenging sequences from the Zurich Mobile Pedestrian corpus generously provided by the authors of [7]. Both sequences, BAHNHOF and SUNNY DAY, were acquired with a stereo rig (13-14fps, 640×480) mounted on a child-stroller. The BAHNHOF sequence was acquired on a crowded sidewalk on a clouded day and contains 999 frames with 5193 annotated pedestrians. The sequence SUNNY DAY was captured on a sunny day and contains 999 frames, 354 of which are annotated with 1867 annotations. For both sequences, there are stereo depth maps, structure-from-motion localization and ground plane estimates available, provided by [7].

Tracking Performance. We use the evaluation criteria from [7]. Tracking quality is measured by the intersection-over-union of tracked person bounding boxes and ground truth annotations in every frame. Matches with an overlap > 0.5 are accepted as correct. Fig. 4(a),(b) presents the performance curves in terms of recall vs. false positives per image (fppi) for different numbers of detection verifications k per frame for both sequences. As can be seen, our approach achieves good performance even when verifying only three ROIs per frame. For comparison, we also provide the curves reported by [7] (only BAHNHOF) and [1, 2, 16] (both sequences). In both cases, our approach achieves higher recall at 0.5 fppi, showing the advantage of depth-based track propagation. At higher precision levels, the performance is only slightly worse than [16], even when only using three detector verifications per frame. The weighting parameters of the Poisson process $w_{\mathtt{tr}}$ and $w_{\mathtt{d}}$ were set to 0.7 and 10.

Furthermore, we evaluated whether modeling the ROI selection with a Poisson process really pays off. To this end, we randomly sampled k ROI candidates in every frame, instead of using the Poisson model, and evaluated them by the detector. As can be seen in Fig. 4(c), the Poisson process model indeed results in better performance (3.5% at 0.5 fppi for five evaluations per frame and 4.8% for three evaluations per frame). The utility factor brings $0.5-1\%$. For a practical application, the benefit is however larger than this number suggests, since the utility factor helps the system focus on tracking close-by persons, which are important for collision avoidance.

In addition, Fig. 4(c) compares our approach's performance to the one of a pure tracking-by-detection system, where the detector slides over the entire image (640x480) and all 27 scales in each frame. The poor performance of the latter can be explained by the fact that we use the same detector setup as in [4] with a 64x128 pixel detection window that constrains the smallest possible detection to this size. However the annotations contain pedestrians that are much smaller than 128 pixels. In contrast, our approach scales ROIs to the appropriate



(a) (b) (c)

Figure 4: Quantitative tracking performance of our approach compared to different baselines on the BAHNHOF and SUNNY DAY sequences from [7].

# ROIs eval. per frame	(all (∅ 12.5))	10	5	3	1	pure tracking by detection 640x480	pure tracking by detection 1280x960
Runtime for 1000 frames (sec.)	101.4	92.4	78.5	64.5	49.6	69.05	220.02
fps	9.86	10.82	12.74	15.50	20.16	14.48	4.54

Table 1: Runtime for the overall system for 1000 frames on the BAHNHOF sequence.

detection size based on the measured distance to the ROI, permitting upscaling of ROIs by up to a factor of 2. This allows us to also detect pedestrians that are farther away from the camera without additional cost. For achieving equivalent performance with a sliding-window detector, we would need to process a 1280×960 image, for which the GPU detector alone requires 180ms per frame (without any tracking), which is far away from the 15fps (Tab. 1).

Computational Performance. The main single computational cost item in a pure tracking-by-detection approach is still the computationally expensive detector. With our approach, we can reduce the detector computation time significantly by attending only to a fixed number of small regions. Evaluating a full 640×480 image with our detector implementation requires 44ms, compared to 2ms for a single region of interest. Overall, our system, including ROI candidate generation, object detection, and tracking runs at more than 15 frames per second (Table 1) on a machine with an Intel Core2 Quad Q9550 @ 2.83GHz, 8GB RAM, and an NVidia GTX 280 graphics card. This does not include stereo computation. However, dedicated hardware solutions [18] and depth sensors (*e.g.* Kinect) are in the meantime available that could take over this job. Also the odometry data is assumed given, which is not a restriction since odometry/SLAM components are standard in mobile robotic systems.

Qualitative Evaluation. Similar to [16], our approach can continue tracking pedestrians that are close to the camera or that are partially occluded by the image boundaries (see Fig. 3). This is an advantage compared to pure tracking-by-detection approaches [7], which cannot continue such tracks robustly due to missing detections. This problem could also be overcome in a pure tracking-by-detection framework by employing a detector with partial occlusion handling, such as the one from [6].

Fig. 3 presents results of our tracker on both test sequences, verifying $k = 5$ regions in each frame. In addition to the tracker bounding boxes, we visualize the depth information that was used for computing the ROIs. As can be seen, our system is able to track most of the visible pedestrians correctly in a very busy environment with many occlusions.

7 Conclusion

We have presented a robust system for mobile street-level multi-person tracking. The core of our system is formed by a stochastic Poisson process that models an optimal ROI candidate

selection given a fixed time budget for the object detector. As our experiments have shown, the approach runs at more than 15 frames per second and reaches state-of-the-art performance, while requiring the verification of only $3-5$ ROIs in every frame. Our results open several interesting research perspectives. By integrating the depth information over time, one can distinguish between moving and static objects and consequently employ Poisson processes with higher rates for moving objects, since those objects are likely to be pedestrians. In future work, we plan to explore multi-class object tracking including cars and bicyclist and to investigate how the Poisson process model could be adapted for different object classes reaching robust tracking performance.

Acknowledgments. This project has been funded, in parts, by the EU project EUROPA (ICT-2008-231888) and the cluster of excellence UMIC (DFG EXC 89).

References

[1] M. Bajracharya, B. Moghaddam, A. Howard, S. Brennan, and L. Matthies. Results from a real-time stereo-based pedestrian detection system on a moving vehicle. In *ICRA*, 2009.

[2] 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.

[3] L. Bombini, P. Cerri, P. Grisleri, S. Scaffardi, and P. Zani. An Evaluation of Monocular Image Stabilization Algorithms for Automotive Applications. In *ITS*, 2006.

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

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

[6] M. Enzweiler, A. Eigenstetter, B. Schiele, and D. M. Gavrila. Multi-Cue Pedestrian Classification with Partial Occlusion Handling. In *CVPR*, 2010.

[7] A. Ess, B. Leibe, K. Schindler, and L. Van Gool. Robust Multi-Person Tracking from a Mobile Platform. *PAMI*, 31(10):1831–1846, 2009.

[8] A. Ess, B. Leibe, K. Schindler, and L. Van Gool. Object Detection and Tracking for Autonomous Navigation in Dynamic Environments. *IJRR*, 29(14):1707–1725, 2010.

[9] S. Frintrop, G. Backer, and E. Rome. Goal-directed Search with a Top-down Modulated Computational Attention System. In *DAGM*, 2005.

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

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

[12] Laurent Itti, Christof Koch, and Ernst Niebur. A Model of Saliency-Based Visual Attention for Rapid Scene Analysis. *PAMI*, 20:1254–1259, 1998.

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

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

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

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

[17] V.A. Prisacariu and I.D. Reid. fastHOG – a Real-Time GPU Implementation of HOG. Technical Report 2310/09, Dept. of Engineering Science, University of Oxford, 2009.

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

[19] P. Dollar S., Belongie, and P. Perona. The fastest pedestrian detector in the west. In *BMVC*, 2010.

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

[21] E. Sommerlade and I. Reid. Information theoretic Active Scene Exploration. In *CVPR*, 2008.

[22] E. Sommerlade and I. Reid. Information-theoretic Decision Making for Exploration of Dynamic Scenes. In *WAPCV*, 2008.

[23] E. Sommerlade and I. Reid. Probabilistic surveillance with multiple active cameras. In *ICRA*, 2010.

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

[25] C. Wojek, G. Dorko, A. Schulz, and B. Schiele. Sliding Windows for Rapid Object Class Localization: A Parallel Technique. In *DAGM*, 2008.

[26] 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.