Object detection and tracking for autonomous navigation in dynamic environments

Andreas Ess · Konrad Schindler · Bastian Leibe · Luc Van Gool

Abstract We address the problem of vision-based navigation in busy inner-city locations, using a stereo rig mounted on a mobile platform. In this scenario semantic information becomes important: rather than modelling moving objects as arbitrary obstacles, they should be categorised and tracked in order to predict their future behaviour. To this end, we combine classical geometric world mapping with object category detection and tracking. Object-category specific detectors serve to find instances of the most important object classes (in our case pedestrians and cars). Based on these detections, multi-object tracking recovers the objects' trajectories, thereby making it possible to predict their future locations, and to employ dynamic path planning. The approach is evaluated on challenging, realistic video sequences recorded at busy inner-city locations.

Keywords object category detection · tracking · dynamic path planning

1 Introduction

Autonomous navigation of robots and cars requires appropriate models of their static and dynamic environment. Remarkable progress has been made in highway traffic situations (Betke et al, 2000) and other largely pedestrian-free scenarios such as the DARPA Urban Challenge (DARPA, 2008). In contrast, environments with a large number of moving agents—in particular inner-city locations with many pedestrians—still pose significant challenges, and autonomous navigation in such circumstances is a largely unsolved problem. One of the main challenges in highly dynamic environments is to predict future states required for decision-making and path planning. We argue that in order to successfully navigate in such scenarios, an environment model needs to include object semantics (*e.g.* whether a moving object is a pedestrian or a car), in order to correctly estimate the objects' motion paths and future locations.

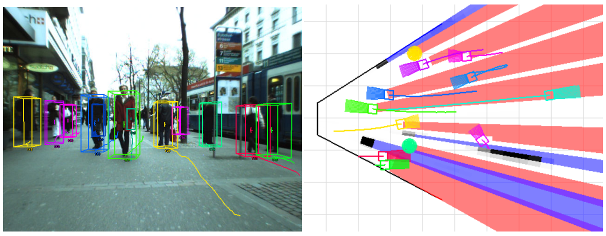


Fig. 1 (left) Navigation in busy urban scenarios requires category knowledge and object tracking, in order to reliably predict future scene states. (right) Overhead view of the scene on the left with detected obstacles (black), tracked persons (coloured boxes with predicted motion cones), and blocked/occluded regions (blue/red shaded areas).

Digital cameras, and in particular binocular stereo rigs, at the moment do not reach the geometric accuracy of range sensors such as LIDAR, but offer the advantage that in addition to the scene geometry they deliver rich appearance information, which is more amenable to semantic interpretation. Recent work has shown that with modern computer vision tools, visual environment modelling for robot navigation is becoming possible (Ess et al, 2009a; Wojek and Schiele, 2008). A key component of these approaches is that they strongly rely on semantic *object category detection*—in the context of road traffic especially detection and tracking of pedestrians and cars.

To support dynamic path planning, it is not sufficient to detect those scene objects; one also has to track them (*i.e.* estimate their trajectories over time) to be able to predict their future locations. The two tasks of detection and tracking are closely related: several of the

A. Ess
Computer Vision Laboratory, ETH Zürich, Switzerland
E-mail: aess@vision.ee.ethz.ch

K. Schindler
Computer Science Department, TU Darmstadt, Germany
E-mail: schindler@cs.tu-darmstadt.de

B. Leibe
UMIC Research Centre, RWTH Aachen, Germany
E-mail: leibe@umic.rwth-aachen.de

L. Van Gool
Computer Vision Laboratory, ETH Zürich, Switzerland *and*
ESAT/PSI–VISICS IBBT, K. U. Leuven, Belgium
E-mail: vangool@vision.ee.ethz.ch

most successful tracking methods at present follow the *tracking-by-detection* paradigm, in which the output of (appearance-based) object detectors serves as observation for tracking. The task of multi-object tracking then amounts to linking the right detections across time to form object trajectories. The approach presented here extends the tracking-by-detection framework to better cope with difficult scenarios with many moving objects close to each other.

The presented system uses only synchronised video from a forward-looking stereo camera pair as input. Based on the video data, the system continuously performs self-localisation, obstacle detection, and object category recognition. The focus of this paper is object tracking and path prediction based on the per-frame output of pedestrian and car detectors. The system delivers all the required input for free-space computation and path planning with dynamic obstacles—see Fig. 1(right).

Key components of our approach are the use of state-of-the-art object class detection to find objects of a given category and of a multi-hypothesis tracker to handle data association in crowded scenes. An important advantage of knowing the object category (semantics) is that one can resort to category-specific, physically meaningful dynamic models for tracking and prediction.

Our focus on purely visual navigation does not preclude the use of other sensors such as LIDAR, GPS/INS, and conventional odometry—in modern robotic systems these sensors have a well-deserved place. Integrating more accurate location and/or geometry could certainly further improve performance of the described system.

The paper is structured as follows: in Sec. 2, we review related work. We then go on to describe the system context in which the tracker is embedded (Sec. 3). In Sec. 4 follows a detailed description of the object tracking method, which is the focus of this paper. Section 5 presents experimental results on five different video sequences, and Sec.6 concludes the paper with a summary and outlook. Parts of the presented work have appeared in the preliminary papers (Ess et al, 2009a,c).

2 Related Work

Tracking as a classical data association problem has been investigated for a long time. Several seminal works originate from the problem of radar tracking (Morefield, 1977; Reid, 1979; Fortmann et al, 1983). These basic works—which are are still in heavy use—already identified the basic ingredients of a tracking system, namely a dynamic (or transition) model describing the objects' motion patterns, an observation (or measurement) model relating the object state to the observed data, and an optimisation algorithm to infer the most likely state from the model. A main advantage of vision as opposed to radar point tracking is the rich appearance information, which can simplify data association in case of close-range targets. On the downside, the observations are often very noisy due to occlusions, lighting conditions, *etc.* Also, practical robotic setups are monocular or binocular with small baseline, which makes localisation in 3D relatively inaccurate. In the following, we will review different tracking approaches under the aspects of the employed dynamic model, observation model, and optimisation strategy. A special focus will be on recent *tracking-by-detection* approaches, which are most similar to the proposed approach.

2.1 Dynamic Model

Although the dynamic model plays a central role in visual tracking, only few models are in common use. In general, one can distinguish between models operating in physically meaningful 3D world coordinates, and models operating directly in the image plane. In the case of tracking in 3D, *e.g.* based on a known ground plane calibration or stereo depth, a constant-velocity model in physical coordinates is the standard choice (Ess et al, 2008; Gavrila and Munder, 2007). When tracking in the image plane, 3D position is often replaced by 2D image position and object scale (Okuma et al, 2004; Wu and Nevatia, 2007; Zhang et al, 2008), but the dynamic models usually remain of first order. Few authors investigated higher-order models for erratic motions, *e.g.* in sports (Okuma et al, 2004), or for interacting targets (Khan et al, 2005). A recent development is to learn image flow fields as dynamic models for densely crowded scenarios (Ali and Shah, 2008). This method does not generalise easily to different scenarios, particularly not to moving cameras, where the flow field constantly changes.

Here we will track in 3D world coordinates with simple physical motion models, namely the constant-velocity model for pedestrians, respectively the Ackermann model (*e.g.* Cameron and Proberdt, 1994) for cars.

2.2 Observation Model

The observation model has evolved considerably over the years. Many early approaches depend on background subtraction (Stauffer and Grimson, 1999; Toyama et al,

1999) followed by blob detection to generate observations. This allows for simple, but general tracking (Isard and MacCormick, 2001; Berclaz et al, 2006; Lanz, 2006), and can be extended to more intricate shape models, *e.g.* (Zhao et al, 2008). A major limitation of background subtraction is the need for a static camera. This constraint is relaxed by using image information such as edges (Isard and Blake, 1998) or local regions (Bibby and Reid, 2008) as observations. However, such low-level structures are susceptible to image clutter. To be more robust to clutter, some researchers have proposed to learn a discriminative model from the low-level features (*e.g.* Avidan, 2005; Grabner and Bischof, 2006).

In recent years, appearance-based object category detection has made great progress, *c.f.* benchmarks such as (Dollar et al, 2009; Everingham et al, 2008; Enzweiler and Gavrila, 2009). As a consequence, using the output of an object detector as observation has become increasingly popular (*e.g.* Okuma et al, 2004; Avidan, 2005; Gavrila and Munder, 2007; Wu and Nevatia, 2007; Zhang et al, 2008; Leibe et al, 2008b). Most trackers operate on the post-processed bounding boxes delivered as a final output of the detector, although some (Leibe et al, 2007b; Ess et al, 2008; Breitenstein et al, 2009) employ a deeper coupling, directly using the (discrete) distribution of detection probabilities. In our experiments, we rely on a popular state-of-the-art object detector (Dalal and Triggs, 2005), but improve its reliability by introducing additional scene context (*c.f.* Sec 3).

In order to have a richer description of the detected object region, the observation is often augmented with local image statistics, mostly colour histograms (*e.g.* Nummiaro et al, 2003; Okuma et al, 2004; Wu and Nevatia, 2007). In recent work, histograms are sometimes replaced by discriminative models of the local texture, which are learnt online. These models are successful in situations where similar objects are close by, especially when trained to discriminate a certain object from all others in the scene (Breitenstein et al, 2009; Song et al, 2008). We found a colour histogram in HSV space to be sufficient for our application, even when simultaneously tracking around 15 people in a busy scenario.

2.3 Optimisation Algorithm

The last required component is an optimisation algorithm, in order to infer the most likely solution under the dynamic and observation models. Many trackers follow the first-order Markov assumption. Under this assumption, the state posterior can be estimated only from the state in the previous frame and the new observation by means of either a (Extended) Kalman filter (Gelb, 1996), Mean-Shift tracking (Comaniciu et al, 2003), the Joint Probabilistic Data Association Filter (JPDAF) (Fortmann et al, 1983), a particle filter (Isard and Blake, 1998), or combinations thereof (Schulz et al, 2001).

These methods however quickly reach their limits as the number of tracked targets increases. To prevent the ensuing combinatorial explosion of the state space, the mentioned filters are typically applied independently to each target and only interact during the data association stage. This interaction mostly amounts to optimising the data assignment per-frame using, *e.g.*, the Hungarian method (Munkres, 1957). Since the state information from all frames but the last one has been discarded (the 1$^{\text{st}}$-order Markov assumption), a very powerful observation model is required to reduce the risk of drifting away from the targets, or at least drifting between targets (switching identities) in busy scenarios.

Drifting can be reduced by optimising data assignment over several time steps. In Multi-Hypothesis Tracking (MHT) (Reid, 1979; Cox, 1993), the k-best assignment algorithm (Murty, 1968) is used in each step to generate the k-best data associations, thus creating a tree of hypotheses, each corresponding to one possible state of the entire scene. This method is quite popular in robotics (Arras et al, 2003). Without careful (heuristic) pruning, MHT quickly becomes prohibitive: for four observed targets and a tree depth of $N = 10$ (easily required to track through short occlusions), $4^{10} \approx 10^6$ state hypotheses have to be handled. Furthermore, by operating only on the data assignment, physical exclusion is not modelled explicitly (although it is to some degree handled implicitly, *e.g.*, by non-maximum suppression of detections in the image plane). Exclusion in the absence of observations is not model-ed, and the tracker can place two objects at the same location. We will refer to such a physically impossible result as a space-time violation.

The combinatorial explosion can be limited by linking detections to "tracklets" where the data association is unambiguous, and optimising data association over those tracklets (Kaucic et al, 2005; Yan et al, 2006; Nillius et al, 2006; Perera et al, 2006; Li et al, 2009). This often gives good results in offline applications, but is not well-suited for online tracking, because in the presence of complex interactions, tracklet generation is only possible in hindsight.

Instead of limiting the state space of associations, Berclaz et al (2006) suggest to discretise the object state (location) to a grid. In this case, the global optimum for a single trajectory can be calculated using the Viterbi algorithm (Viterbi, 1967). The extension to multiple targets is done in a greedy fashion, iteratively running the algorithm and removing the most probable candidate, which turns out to be a heuristic for the exact algorithm defined in Wolf et al (1989). For mobile applications, defining a grid is difficult and the discretisation can cause aliasing problems during path prediction.

Recently, some authors have suggested the use of Markov chain Monte Carlo (MCMC) sampling to find the (approximate) optimum in the joint tracking space (Khan et al, 2005; Yu et al, 2007; Zhao et al, 2008). This method generates a sequence of states at a given time step, which collectively approximate the target distribution. It can be seen as a stochastic version of the hypothesise-and-verify strategy employed in the present work.

Jiang et al (2007) propose the use of Integer Linear Programming (ILP, Schrijver, 1998) to find a globally consistent solution, accounting for exclusion and occlusion between objects. This method can be employed online, but is susceptible to changes of the observed bounding box, and needs to know the number of objects in the scene *a priori*.

An elegant graph-theoretic solution to the data association problem was proposed in (Zhang et al, 2008). Each detection is represented by two nodes in a graph, and edges are used to model transition, enter, and exit probabilities, respectively. The globally optimal assignment of detections to trajectories is then obtained by repeatedly running a min-flow algorithm with different flows, where the flows intuitively correspond to the number of people in the scene. Explicit occlusion reasoning is possible in a second step, by adding virtual occluded detections to the graph. As above, space-time violations are only handled implicitly, and for each missing detection extra edges need to be added to the graph, which can become prohibitive in the case of prolonged misses. The method is more suited for offline application, since it does not *per se* generate an output for objects not observed in the current frame.

Morefield (1977) was probably the first to suggest tracking by means of a two-stage hypothesise-and-verify procedure. An over-complete set of trajectory *candidates* is generated from the entire batch of measurements over all time steps. ILP is then employed to find a consistent subset, under the constraint that each data point (*i.e.* detection) is used only once.

Independently, Leibe et al (2007a) introduced a tracking framework in which a redundant set of candidate trajectories is generated online in each time step and is pruned to a consistent subset using statistical model selection. To account for missing measurements, interactions are not only modelled by penalising the repeated use of the same detection, but also the simultaneous occupation of space-time volume. The present work builds on this framework, with a focus on trajectory estimation and prediction for autonomous outdoor navigation.

3 Background

3.1 Object Category Detection

Appearance-based object detection forms the backbone of our approach. In this paper, we focus on the two most relevant object categories for street scenes: *pedestrians* and *cars*. Detection delivers two important pieces of information, namely *where* in the image an object of interest is located, and *what type of object* it is. The former serves to establish the objects' positions in the world, which in a dynamic setting are an important part of the geometric environment model. The latter provides semantic information that supports higher-level reasoning—most importantly for choosing the right motion model for the object, and for taking navigation decisions which depend on the object category (*e.g.* the safe distance to pass a pedestrian may be different from the one to pass a car).

In more detail, the required detector output for further probabilistic treatment is a set of potential object detections, each consisting of an image position, scale, object category label, and a *detection probability* $p(o_i|\mathcal{I})$. In the present work we employ the popular HOG framework (Dalal and Triggs, 2005). In a nutshell, this method uses a binary support vector machine (SVM) to classify an image region as object or non-object. As input features, it extracts a robust gradient-based encoding of the image structure, the "Histogram of Oriented Gradients" (hence the name HOG). The classifier is evaluated densely at each image position and over a multi-scale pyramid. The output is then converted to a large set of potential detections by extracting all local maxima of the detection probability which exceed a low threshold. Their SVM scores can easily be transformed to yield the desired detection probabilities.

For our application the simple combination of a global region descriptor and a linear classifier consistently outperforms other popular detection frameworks such as the *implicit shape model* (Leibe et al, 2008a) and a deformable part-based model (Felzenszwalb et al, 2008)—a detailed comparison is given in (Ess et al, 2009b). The reason seems to be that pedestrians walking on the street, respectively cars, exhibit relatively small shape variations, so that the additional flexibility of such models is outweighed by their lower discriminative power.

For pedestrians, a single detector proved to be sufficient, whereas seven separate detectors were trained for different views of cars, (three for *front-profile, side, and back-profile* views, each in two mirrored version for the left and right side, and one for *back* views). In our approach the desired output of the classifier is a probability that a given object class is present at a certain location, which then serves as input for the subsequent hypothesis generation and selection. Since that step explicitly enforces physical exclusion and resolves cases where multiple objects would occur at the same spatial location, we opt to run independent binary classifiers for each category and viewpoint. Postponing the competition between different objects to a later stage allows one to resolve it in 3D space, and also circumvents the problems associated with multi-class classification such as calibrating the margins returned by different classifiers, and comparing image windows of varying aspect ratio.

3.2 World Coordinate System

To allow reasoning about object trajectories in 3D coordinates, the camera position in the world coordinate system is estimated at each frame ("visual odometry", Nistér et al, 2004). Compared to standard visual odometry, our system includes scene knowledge obtained from the tracker to mask out image regions which do not show the static background. Furthermore, our system explicitly detects failures by comparing the estimated position to a Kalman filter prediction. In the event of failure, the visual odometry is re-initialised to yield collision-free navigation (at the cost of possible global drift). For details, we refer to (Ess et al, 2009b).

3.3 Ground Plane Estimation

Instead of directly using the output of a pedestrian detector for the tracking stage, we introduce a simple scene model: all objects of interest are assumed to reside on a common ground plane. This ground plane is not fixed, but varies smoothly, so as to allow for hilly terrain as well as camera tilt (*e.g.* due to acceleration/deceleration of the robotic platform).

The ground plane and the set of detected pedestrians are jointly estimated in a Bayesian network, using as input detection probabilities from the object detection stage and stereo depth, as well as priors on object size and a temporal smoothness prior on the ground plane. Joint estimation has the advantage that evidence is propagated in both directions: for largely empty scenes the ground plane can be reliably estimated from depth measurements and significantly constrains object detection; in crowded scenes less ground is visible, but a large number of detected objects may in turn constrain the ground plane.

Compared to the object detector alone, the combination with the ground plane model, stereo depth, and object size priors greatly reduces the number of false positives. For details, see (Ess et al, 2007).

3.4 Static Obstacles

For obstacles other than the detected object categories *pedestrian* and *car*, in particular static street furniture, additionally we construct a stochastic occupancy map with the method of (Badino et al, 2007): depth maps are projected onto a polar grid on the ground and are integrated over time to yield an obstacle map. In contrast to the original method, we filter out the tracked scene objects. This is done for two reasons. Firstly, integrating moving objects results in smeared occupancy maps. Secondly, we are interested not so much in the *current* positions of the pedestrians and cars as in their *future* locations. These can be predicted more accurately with a specific motion model inferred from the tracker.

4 Tracking

Given the information described in the previous section, tracking amounts to fitting a set of trajectories to the potentially detected objects in 3D world coordinates, such that these trajectories together have a high posterior probability, *i.e.*, they explain the observed evidence well. Unlike traditional Markovian trackers, our approach applies a hypothesise-and-verify strategy in order to find the set of trajectories that best explains the evidence from past and present frames. The *hypothesise* step samples a large, over-complete set of candidate trajectories, and the *verify* step prunes it to a minimal consistent subset.

The basic units of a tracker in the hypothesise-and-verify framework are hypotheses (candidates) for possible object trajectories. Such a trajectory hypothesis is defined as $H_j = [\mathcal{S}_j, \mathcal{M}_j, \mathcal{A}_j]$, where $\mathcal{S}_j$ denotes its supporting detections, $\mathcal{M}_j$ its dynamic model, and $\mathcal{A}_j$ its appearance model. The set of all candidate hypotheses at time t is denoted $\mathcal{H}^t_{cand}$. This set can be (and usually will be) redundant, with many spurious hypotheses. The verification stage then selects the optimal subset of trajectories $\mathcal{H}^t_{sel}$.

The basis for tracking-by-detection are the detections $o_i^{t_i} = [\mathbf{x}_i, \mathtt{C}_i, t_i, a_i]$, with $\mathbf{x}_i$ the object's 2D position on the ground plane, t_i the time-stamp (frame index), $\mathtt{C}_i$ the covariance matrix capturing the positional uncertainty, and a_i the appearance. Based on the output of object detection and ground plane estimation (Sec. 3) in a frame t_i, we denote by $p(o_i^{t_i}|\mathcal{I}^{t_i})$ an object's probability given the image evidence, *i.e.* its detection score and the ground plane. For the sake of clarity we will mostly omit the superscript t_i in the following. The detections are accumulated in a space-time volume $\mathcal{O}$ that spans all previous frames up to the current one. To keep the method computationally tractable, $\mathcal{O}$ in practice only contains the last few hundred time steps, with t_0 the smallest time step still considered. The aim of the tracking step is thus to fit smooth trajectories H_j to the detected object locations $[\mathbf{x}_i, t_i]^\top$ in $\mathcal{O}$.

4.1 Hypothesis generation

The set of candidate trajectories is generated by running bi-directional Extended Kalman Filters (EKFs) starting from each detection within $\mathcal{O}$ (for computational efficiency, the candidates from previous frames are cached and extended, and only those starting from new detections are generated from scratch, see below). Each filter generates a candidate trajectory which obeys the physical motion constraints of a person or car, respectively, and bridges short temporal gaps due to occlusion or detection failure. Note that candidates do *not* only originate from the accepted tracks of the last frame (as in classical trackers built on a first-order Markov assumption). In the following, we describe the hypothesis generator in more detail.

4.1.1 Data Association

In order to reliably associate a trajectory hypothesis with candidate detections, we employ for each hypothesis H_j both a dynamic model $\mathcal{M}_j$ and an appearance model $\mathcal{A}_j$. Together, these can be used to evaluate an observation o_i under H_j,

$$p(o_i|H_j) = p(o_i|\mathcal{A}_j) \cdot p(o_i|\mathcal{M}_j) \ . \tag{1}$$

The probability $p(o_i|H_j)$ is used to score all observations of a time-step against a trajectory hypothesis. The detection with the highest probability is then used for updating the trajectory ("winner takes all"). To prevent erroneous associations, $p(o_i|H_j)$ is gated so as to include only feasible observations.

Dynamic Model. Following a standard approach, we use an Extended Kalman Filter (Gelb, 1996) to describe an object's motion in a physically plausible way. In the following, we briefly review the generic model, before describing the actual motion models for our object categories.

An EKF is a recursive Bayesian filter (for a general introduction, see *e.g.* Arulampalam et al, 2002; Gelb, 1996) which iteratively repeats two steps at each frame: it *predicts* the object state by applying the dynamic model $\mathcal{M}$ to the state posterior s_{t-1} of the previous frame; and it *updates* the resulting state prior to a state posterior s_t for the current frame by fusing it with the new observation o_i. The dynamic model $\mathcal{M}$ gives rise to a function $f^{\mathcal{M}}(\cdot)$, which governs the state transition $p(s_t|s_{t-1})$. Assuming a first-order model, the *a priori* distribution of the next time step can be calculated given measurements up to time $t-1$ via the Chapman-Kolmogorov equation,

$$p(s_t|O_{t-1}) = \int p(s_t|s_{t-1})\, p(s_{t-1}|O_{t-1})\, ds_{t-1} \quad , \tag{2}$$

where $O_t = \{o_1, \ldots, o_t\}$ denotes the set of observations up to time t. Taking a new measurement o_t into account, the predicted distribution is updated according to Bayes' rule to arrive at the *a posteriori* distribution

$$p(s_t|O_t) = \frac{p(o_t|s_t) \cdot p(s_t|O_{t-1})}{p(o_t|O_{t-1})} \quad , \tag{3}$$

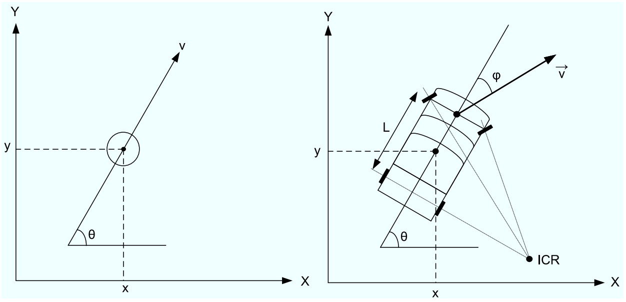
Here $p(o_t|O_{t-1})$ is a normalisation factor and $p(o_t|s_t)$ is the observation likelihood (the likelihood that state s_t generated measurement o_t).

Due to the large state space for multi-dimensional state vectors, the evaluation of the prior probability for each point quickly becomes intractable. Our EKF framework, an extension of linear Kalman filtering, assumes a unimodal Gaussian distribution of the current state. It is specified by defining the transition function $f^{\mathcal{M}}(\cdot)$ and the measurement function $f^{\mathcal{X}}(\cdot)$ (for pedestrians the observed location, for cars the observed location and heading direction), as well as their respective Jacobians.

A more general recursive Bayesian filter, which caters for multi-modality, is the particle filter (sequential Monte-Carlo estimation). In our experiments using a particle filter did not yield any improvement. It behaved similar to the EKF, even when observations were missing (*i.e.*, a radially growing uncertainty ellipse).

In the employed EKF framework, motion models only differ in the choice of the state transition function $f^{\mathcal{M}}(\cdot)$ and its noise vector. In the following, we will introduce the models used for pedestrians and cars.

Pedestrians. For pedestrians, we assume a constant-velocity model, *i.e.*, the state space is defined as $\mathbf{s}_t = [x_t, y_t, \theta_t, v_t]^\top$, with (x_t, y_t) the 2D position, θ_t the pedestrian's orientation, and v_t its speed, see Fig. 2 (a). The latter two are initialised to 0, as a detection itself only



(a) (b)

Fig. 2 (a) Constant-velocity model employed for pedestrians. (b) Ackermann steering model used for describing a car's non-holonomic motion. ICR denotes the centre of rotation, tangential to which the acceleration is measured.

indicates the position of the person. The corresponding transition function is thus

$$f^{\mathcal{M}}(\mathbf{s}_{t-1}, w_{t-1}) = \begin{pmatrix} x_{t-1} + v_{t-1}\cos(\theta_{t-1})\Delta t \\ y_{t-1} + v_{t-1}\sin(\theta_{t-1})\Delta t \\ \theta_{k-1} \\ v_{t-1} \end{pmatrix} + \begin{pmatrix} 0 \\ 0 \\ w_\theta \\ w_v \end{pmatrix} . \quad (4)$$

w_θ and w_v are the noise components for orientation and velocity, respectively. Given the current state position $\mathbf{x}_t^s$, the likelihood of an object o_i with position $\mathbf{x}_i$ under the motion model is measured by:

$$p(o_i|\mathcal{M}_j) \sim e^{-\frac{1}{2}(\mathbf{x}_i-\mathbf{x}_t^s)^\top(\mathtt{C}_t+\mathtt{C}_{x_i})^{-1}(\mathbf{x}_i-\mathbf{x}_t^s)} \quad , \quad (5)$$

This accounts for the uncertainty in the system $\mathtt{C}_t$, as well as for the localisation uncertainty of the detection $\mathtt{C}_{x_i}$, estimated from the stereo geometry by linear error propagation. The latter is especially important to handle far away objects correctly, as their localisation in depth is highly inaccurate. Correct modelling of these two terms proved to be of prime importance for good tracking results across a large working range.

Cars. For cars, which move non-holonomically due to mechanical constraints, we employ the Ackermann steering model (*c.f.* Cameron and Proberdt, 1994). This includes two so-called *driving processes*, the steering angle ϕ_t and the tangential acceleration a_t, as shown in Fig. 2(b). The state vector is $\mathbf{s}_t = [x_t, y_t, \theta_t, v_t, \phi_t, a_t]^\top$, giving rise to the update equation

$$f^{\mathcal{M}}(\mathbf{s}_{t-1}, w_{t-1}) =$$
$$\begin{pmatrix} x_{t-1} + v_{t-1}\cos(\theta_{t-1})\Delta t + \frac{1}{2}a_{t-1}\cos(\theta_{t-1})\Delta t^2 \\ y_{t-1} + v_{t-1}\sin(\theta_{t-1})\Delta t + \frac{1}{2}a_{t-1}\sin(\theta_{t-1})\Delta t^2 \\ \theta_{t-1} + \frac{1}{L}v_{t-1}\tan(\phi_{t-1}) \cdot \Delta t \\ v_{t-1} + a_{t-1} \cdot \Delta t \\ \phi_{k-1} \\ a_{t-1} \end{pmatrix} + \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \\ w_\phi \\ w_a \end{pmatrix} . \quad (6)$$

L is the distance between the axles of the car and is set to a default value of $L = 3.2\,\mathrm{m}$ in our scenario. The process noise (w_ϕ, w_a) is only dependent on the two driving processes.

Note that the EKF cannot account for any constraints on the steering angle. We nevertheless chose this option for its simplicity and good performance. Cars are detected with separate detectors for different viewpoints, therefore the likelihood function is extended to also take into account the orientation $\widehat{\theta}_i$ of the detector:

$$p(o_i|\mathcal{M}_j) \sim e^{-\frac{1}{2}(\mathbf{x}_i-\mathbf{x}_t^s)^\top(\mathtt{C}_t+\mathtt{C}_{x_i})^{-1}(\mathbf{x}_i-\mathbf{x}_t^s)-\lambda|\mathbf{n}(\theta_t^s)^\top\mathbf{n}(\widehat{\theta}_i)|} \quad , \quad (7)$$

where λ sets the influence of the orientation similarity on the distance and $\mathbf{n}(\theta)$ denotes the unit vector corresponding to the orientation θ. The employed motion models are rather simple, but proved to be effective in practice. In particular, we found that the positional accuracy of the underlying detections is not sufficient to support more complex motion models (*e.g.* incorporating acceleration).

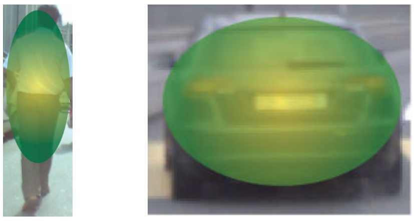


Fig. 3 Colour histograms are calculated inside an ellipse fitted to the bounding box, weighted with a Gaussian.

Appearance Model. As a hypothesis H_j's appearance (observation) model $\mathcal{A}_j$, we choose an $(8 \times 8 \times 8)$-bin colour histogram in HSV space. For each observation o_i, we compute its histogram a_i over an ellipse fitted inside the detected bounding box, applying a Gaussian kernel to put more emphasis on pixels close to the centre. The resulting ellipses for pedestrians and cars are shown in Fig. 3. For robustness against slight colour aberrations, trilinear interpolation is used when building the histograms.

The similarity of an object and a hypothesis is then defined by the Bhattacharyya distance between the histograms,

$$p(o_i|\mathcal{A}_j) \sim \sum_q \sqrt{a_i(q)\mathcal{A}_j(q)} \, , \quad (8)$$

with q a 3-dimensional index over the histogram bins.

When a new observation o_i is added to a trajectory, $\mathcal{A}_j$ is updated with an Infinite Impulse Response (IIR) filter,

$$\mathcal{A}_j(q) = w\mathcal{A}_j(q) + (1-w)a_i(q) \quad . \quad (9)$$

The mixing factor $w = 1 - \min\big(\exp(|\mathbf{x}_i - \mathbf{x}_t^s|^2), \gamma\big)$ depends on the closeness of the observation and the dynamic model, so as to reduce the influence of badly localised detections which contain more pixels from the background and fewer pixels from the object. To limit the influence of a single observation and ensure smoothly varying appearance, the mixing factor is also truncated at $\gamma = 0.9$.

The appearance model is only used to rule out very improbable associations, but is not propagated through the Kalman filter, which would prohibitively inflate the state vector. In combination with the dynamic model, this captures enough information to allow reliable tracking among many interacting agents. Further improvements could potentially be obtained by using a discriminatively learnt model (Song et al, 2008), where an object's appearance is explicitly learnt against all the others in an online fashion. Note that the discriminative approach cannot be applied when trajectory hypotheses are generated independently, as in (Leibe et al, 2008b). Such an approach requires the knowledge of all tracked objects (*i.e.* not *candidates*) at each time step, to compare classifier confidences.

Efficiency Considerations. The space-time volume $\mathcal{O}$ accumulates detection responses from the current and past frames and serves as the basis for creating trajectory hypotheses. The employed hypothesise-and-verify architecture requires that the set of hypotheses passed to the selection algorithm has to be over-complete, *i.e.* the hypothesis generation stage has to generate *at least* all correct hypotheses for the subsequent selection to recover an object. However, to keep the method computationally tractable, the candidate set should also be as small as possible, as the general optimisation procedure is NP-complete. Therefore, we generate hypotheses drawing inspiration from two complementary approaches in the literature. On the one hand "trajectory extension" is very similar to a standard Markovian tracker, where in each step hypotheses compete for evidence (*e.g.* Wu and Nevatia, 2007). We thus expect a performance at least as good as a Markovian tracker. On the other hand, we start searches backwards in time to initialise hypotheses and generate possible additional explanations, which can be interpreted as an instance of the "observe-and-explain" approach (Ryoo and Aggarwal, 2008).

4.1.2 Trajectory Generation and Extension

Conceptually, trajectories can be initialised at each potential detection in the volume $\mathcal{O}$ by applying the appropriate motion model backwards and forwards in time. This algorithm is visualised in Fig. 4 (a–c).

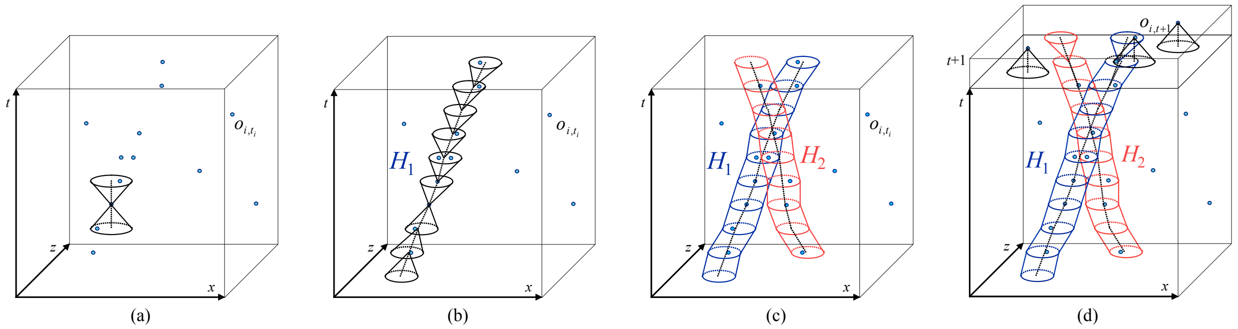
In typical scenes, this creates a candidate set with many unnecessary duplicates started at different detections on the same track. To prevent these, the hypothesis set can be constructed in an incremental fashion by (1) extending hypotheses forward in time from the last time step to the current one, and (2) starting independent searches backwards in time from all detections of the current time step. The latter ensures that trajectories for newly appearing objects are initialised, and that for each actual object which has been successfully detected in the current frame, at least one good trajectory candidate is created, independent of earlier association errors.

To maximise tracking performance, it is crucial to find (among others) good candidates with correct data association. This requires some care in handling ambiguities in the association process and occlusions.

Parallel Generation. Trajectory generation is performed in parallel from all new detections (in contrast to (Leibe et al, 2008b)). When generating candidate trajectories independently of each other, they cannot compete for measurements—the competition is left to the final selection algorithm. In difficult crowded cases, candidates will therefore include wrong measurements of other nearby objects. To remedy this behaviour, we rely on the fact that image-based non-maximum suppression only yields at most one detection per object and camera, and that the observations of the same object in two or more cameras can easily be merged with a conservative clustering step. Hence, only the most likely detection is used to update the state ("winner-takes-all"), rather than using all nearby detections weighted by the distance.

To sustain existing hypotheses already in the hypothesis set, we extend them, similar to standard recursive trackers. Again, the extension is carried out in parallel such that trajectories compete for detections, ensuring that each trajectory can be updated with at most one observation. In case of conflicts, when an observation is the most likely match for two or more candidate trajectories, the observation is assigned to the trajectory candidate with the highest likelihood, in a greedy manner. Candidates which do not manage to claim any detection during this process are merely extended through extrapolation.

The effect of the competitive hard assignment of detections is twofold. Firstly, it avoids unwanted attraction between candidates and better separates closely interacting pedestrians (when using soft assignment, the same measurement can influence several nearby trajectory candidates, pulling them closer together). Secondly, the set of candidates tends to be more compact, because each measurement can only support a single



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

Fig. 4 Illustration of the trajectory growing procedure. (a) Starting from an object detection at some point in time, detections in neighbouring frames are found which can be reached within the constraints of the dynamic model. (b) The trajectory is adapted based on the new observations, and the process is iterated both forwards and backwards in time. (c) This results in a set of candidate trajectories, which are passed to the hypothesis selection stage. (d) For efficiency reasons, trajectories are grown incrementally.

candidate in a crowded region, making weak candidates more prone to attrition.

Occlusion Reasoning. The limited height of most vehicles enforces rather low camera placement, such that pedestrians and/or cars are frequently occluded by each other, or by other scene objects. We therefore opt to explicitly model occlusion, rather than to treat it as yet another case of missing detections. To this end, we generate an occlusion map on the ground plane, on the same discrete polar grid as the obstacle map (*c.f.* Sec. 3). An example is shown in Fig. 1. The map contains the regions occluded by both static obstacles (occlusions computed directly from the obstacle map) and moving objects (occlusions computed from object positions extrapolated from the previous time step).

As long as a candidate trajectory remains in an occluded region, it is kept alive and its state is extrapolated. Here the uncertainty modelling property of the EKF becomes important: prolonged extrapolation without observations leads to progressively larger location uncertainty and hence a larger search region for supporting detections. This increases the chance of finding the observation again once the object reappears. The *greedy assignment* described in Section 4.1.2 meanwhile ensures that such a candidate does not steal detections from less uncertain competitors. By modelling occlusions, we obtain longer object tracks, which better supports path planning.

4.2 Trajectory selection

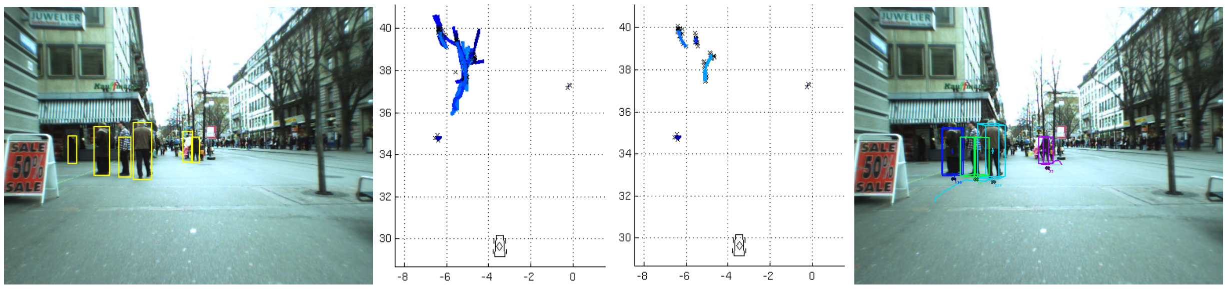
The obtained candidate set $\mathcal{H}_{cand}$ is highly redundant, and the candidate trajectories are not independent because of the twin constraints that two pedestrians cannot occupy the same location on the ground plane at the same time and that each object detection can only belong to a single pedestrian trajectory.

The selection step employs model selection to discard all redundant hypotheses and retain only a minimal, conflict-free set of trajectories required to adequately explain past and present observations. This is achieved by maximising the total support of the pruned set $\mathcal{H}_{sel}$ as a function of the selected candidates. This support should

– increase as the trajectories $\mathcal{H}_{sel}$ explain more detections and as they better fit the detections' 3D location and 2D appearance through the individual contribution of each detection;
– decrease when trajectories are (partially) based on the same object detections, through pairwise corrections to the trajectories' joint support (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);
– decrease with the number of required trajectories through a prior favouring explanations with fewer trajectories—balancing the model complexity against its goodness-of-fit in order to avoid over-fitting ("Occam's razor").

Fig. 5 visualises the generation and selection of candidate trajectories for an example scene. In this scene, people are standing close together, which results in trajectory hypotheses that contain detections from several actual persons (note, *e.g.*, the long curve going to the left). Selecting such a candidate is however suboptimal from a global perspective, as the above-mentioned constraints would preclude the simultaneous selection of other candidates that are based on the same persons. Hence, it is better to select candidates that are mutually consistent with each other.

Note that starting from an exhaustive set of trajectory candidates by definition enables automatic initialisation (usually after 2–3 detections) and the ability to recover from temporary track loss and occlusion.



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

Fig. 5 Tracking by means of a hypothesise-and-test framework: given object detections from the current and past frames (a), we construct an exhaustive, over-complete set of trajectory hypotheses (b) and prune it back to an optimal subset with model selection (c), yielding the final trajectories (d).

To select the jointly optimal subset of trajectories, we assign each trajectory H_j a support (utility) $\mathcal{U}$, which is composed of the strength of its supporting detections $\{o_i\}$, weighted by their goodness-of-fit w.r.t. the dynamic model $\mathcal{M}$ and the appearance model $\mathcal{A}$.

$$\begin{aligned}\mathcal{U}(H_j|\mathcal{I}^{t_0:t}) &= \sum_i \mathcal{U}(o_i|H_j,\mathcal{I}^{t_i}) = \\ &= \sum_i p(o_i|\mathcal{I}^{t_i}) \cdot p(o_i|\mathcal{A}_j) \cdot p(o_i|\mathcal{M}_j)\ .\end{aligned} \qquad (10)$$

Choosing the best subset $\{H_j\}$ is now a model selection task. If we only take into account pairwise interactions[1] it translates to the quadratic binary problem

$$\max_{\mathbf{m}} \left[\mathcal{D}(\mathbf{m})\right] = \max_{\mathbf{m}} \left[\mathbf{m}^\top \mathbf{Q}\mathbf{m}\right] \quad , \quad \mathbf{m} \in \{0,1\}^N\ . \qquad (11)$$

Here $\mathbf{m}$ is an indicator vector (of length N), specifying which candidates to use ($m_i = 1$) and which to discard ($m_i = 0$). The diagonal elements q_{ii} contain the individual likelihoods of candidate trajectory H_i, reduced by the “model penalty”, a prior which favours solutions with few trajectories. The off-diagonal elements q_{ij} model the interaction between candidates i and j and contain the correction for double-counting detections consistent with both candidates, as well as a penalty proportional to the overlap of the two trajectories' footprints on the ground plane:

$$\begin{aligned} q_{ii} =& -\epsilon_1 + \sum_{o_k^{t_k} \in H_i} \Big((1-\epsilon_2) + \epsilon_2 \mathcal{U}(o_i^{t_i}|H_j,\mathcal{I}^{t_i})\Big) \\ q_{ij} =& -\frac{1}{2}\epsilon_3 O(H_i,H_j) - \\ & -\frac{1}{2}\sum_{o_k^{t_k} \in H_i \cap H_j} \Big((1-\epsilon_2) + \epsilon_2 \mathcal{U}(o_i^{t_i}|H_\ell,\mathcal{I}^{t_i})\Big)\ , \end{aligned} \qquad (12)$$

where $H_\ell \in \{H_i, H_j\}$ denotes the weaker of the two trajectory hypotheses, whose evidence is subtracted to avoid double counting; $O(H_i, H_j)$ measures the physical overlap between the footprints of H_i and H_j given average object dimensions; ϵ_1 is the base cost for each new trajectory, required to prevent over-fitting, and should be chosen such that it suppresses trajectories with less than ≈ 2 good detections, in order to weed out erratic false detections; ϵ_2 is a regularisation parameter, which ensures a minimal support for each explained object detection and compensates for model inaccuracies—smaller ϵ_2 puts less weight on the goodness-of-fit in terms of appearance and dynamics, and more weight on the fact that a detection could be associated with the trajectory at all; ϵ_3 is the influence weight of the overlap penalty, and should be chosen large enough to prevent selecting any two trajectories with significant overlap.

4.2.1 Optimisation

The maximisation of Eq. (11) is NP-hard, but there are several methods which find strong local maxima, *e.g.* the multi-branch method of (Schindler et al, 2006), or QBPO-I (Rother et al, 2007). The solution is a locally optimal set of object candidates for the current frame: most false detections are weeded out, since they usually do not have a supporting trajectory in the past (this is the main source of improvement), and missed detections are filled in by extrapolating those trajectories which have strong enough support in the previous frames.

In our approach, we use an extended version of the multi-branch method of (Schindler et al, 2006). The algorithm selects at every level the set of most promising hypotheses and tries adding each one recursively. At each level R, at most B_R hypotheses are tested. An important insight of Schindler et al (2006) is that the function $\mathcal{D}$ is submodular ($q_{ii} > 0$, and $q_{ij} \leq 0\ \forall i \neq j$). Due to this property, the path to the optimum never

[1] Disregarding higher-order interactions results in too high penalties in cases where more than two trajectories compete for the space and/or detections; if interaction penalties are high enough to enforce complete exclusion, this will not alter the result.

contains any descent steps. Formally, given a current choice $\mathbf{m}'$, the next step in the search $\mathbf{m}''$ must always fulfil $\mathcal{D}(\mathbf{m}'') > \mathcal{D}(\mathbf{m}')$. Otherwise, the search along this path can be terminated.

Here, we employ an additional bound. Given a solution $\mathbf{m}'$, let $\mathcal{L}'$ be the set of indices of hypotheses which are *not* in the current solution, $\{\forall i \in \mathcal{L}' : m'_i = 0\}$. Denote by $\mathbf{1}_i$ a vector that contains all 0s except at entry i. Starting from $\mathbf{m}'$, the maximally reachable score never exceeds

$$s = \mathcal{D}(\mathbf{m}') + \max \Big[0, \sum_{i \in \mathcal{L}'} \left(\mathcal{D}(\mathbf{m}' + \mathbf{1}_i) - \mathcal{D}(\mathbf{m}') \right) \Big] . \quad (13)$$

That is, the maximal additional score is always bounded by the sum of adding all positive hypotheses, accounting for their interaction with the current solution, but ignoring their interactions among each other. Formally[2], $\mathcal{D}(\mathbf{m}' + \mathbf{1}_i + \mathbf{1}_j) + \mathcal{D}(\mathbf{m}') \leq \mathcal{D}(\mathbf{m}' + \mathbf{1}_i) + \mathcal{D}(\mathbf{m} + \mathbf{1}_j)$. As a consequence of this bound, the optimisation can quit a branch of the search tree as soon as the maximally reachable score is smaller than the current maximum s_{max}. This early stopping reduces the number of invocations of the search recursion to 63% of the original number. When operating on a video sequence, the solution from the previous frame can be used to further constrain the optimisation. To do so, we set $s_{max} = \mathcal{D}(\hat{\mathbf{m}}^{(t-1)})$ before starting the optimisation, where $\hat{\mathbf{m}}^{(t-1)}$ are the trajectories deemed successful in the last time step, evaluated under the *current* interaction matrix $\mathbf{Q}$. Doing so ensures that s_{max} is always a reachable solution. Due to temporal consistency, this is usually a good starting value for s_{max}, reducing the number of invocations by another 2% to 61% of the original number (which is only a small improvement, but comes at no extra cost).

Discussion. As the maximisation is performed on a per-frame basis, there is no guarantee that the current explanation is consistent with the one obtained in the previous frame. Still, when it is selected it not only explains the current frame t, but also offers the most likely explanation for the past, in the light of the entire evidence up to time t. We can thus follow a trajectory back in time to determine where a pedestrian or car came from when it first entered the field of view, even if back then no trajectory was selected for that particular object.

Empirically, the selection step typically keeps between 25% and 35% of the candidate trajectories. In extreme cases, this figure extends to 8% and 100%, respectively. The ratio depends on the momentary complexity of the scene: pedestrians with similar appearance moving close to each other give rise to more candidates. Such difficult situations are also the cases where greedy maximisation of Eq. (11) fails. With the proposed optimisation method, we did not notice any problems due to weak minima. The limiting factor seems to be model inaccuracy, rather than optimisation failures.

4.3 Implementation Issues

In this section, we review some important details of the practical implementation. Although these details are mainly straight-forward engineering considerations, we discuss them in some detail, in the hope that they may be useful for other researchers.

Hypothesis Pruning. Continually extending the existing hypotheses and at the same time generating new ones leads to an ever-growing hypothesis set, which would quickly become intractable. A conservative pruning procedure is used to control the number of hypotheses to be evaluated: (1) hypotheses older than the time window under observation (t_0–t) are removed, (2) candidates which have been extrapolated through time for too long without finding any new evidence are removed, and (3) candidates which have been in the hypothesis set for too long without having ever been selected are discontinued (these are mostly weaker hypotheses, which are always outmatched by others in the competition for space).

Importantly, the pruning step only removes hypotheses which have been unsuccessful over a long period of time. All other hypotheses, including those not selected in recent frames, are still propagated and are thus given a chance to find new support at a later point in time. This allows the tracker to recover from failure and retrospectively correct tracking errors.

4.3.1 Identity Management

While the hypothesis selection framework uses information from a large time interval, the resulting explanations are independent at each time step. Thus, object identities are not automatically preserved. If this is desired (*e.g.* for surveillance scenarios), an additional process is needed to propagate identities.

In the case of a trajectory generated by extension of a previously selected one, identity preservation is trivial. If the selected trajectory H_j is not the extension of a previously successful candidate, it is compared to the "old" trajectories selected in past frames (which already have a unique identity). If an "old" trajectory is

[2] This follows directly from the definition of submodularity (*c.f.* Boros and Hammer, 2002).

found, which is based largely on the same detections $\mathcal{S}_k$, then the new trajectory is assigned the identity of the old one. If the new trajectory does not match any of the known trajectories, a new ID is instantiated. As a criterion for trajectory overlap we use

$$|\mathcal{S}_j \cap \mathcal{S}_k| / \min(|\mathcal{S}_j|, |\mathcal{S}_k|) > \xi \, . \tag{14}$$

By itself this may seem like a crude heuristic. However in the context of the presented system, we can choose a very conservative threshold ξ (say, 50%), because the physical exclusion constraints during trajectory generation and selection ensure that any two trajectories selected at time t have zero overlap (and hence only one trajectory at any time t can significantly overlap a reference trajectory from a previous time step).

4.3.2 Trajectory Initialisation and Termination

Tracking is started automatically after a few frames as soon as the benefit of a correct trajectory exceeds its cost. The initialisation is not constrained to a specific image region, since such "entry regions" cannot be defined for general scenarios, even less so if the camera is moving. Although several frames are required as evidence for a new track (in our application 2–3, given evidence from both cameras), the trajectory is in hindsight recovered from its beginning.

The flipside of automatic initialisation is that trajectory termination must be handled explicitly. If an object leaves the scene, the past detections along its track still exist and may prompt unwanted re-initialisations. To avoid this behaviour, exit zones are defined on the ground plane along the image borders[3] and are constantly monitored. When a trajectory enters the exit zone from inside the image, the corresponding object is labelled as terminated, and its final trajectory is stored in a list of terminated tracks. During hypothesis selection, these terminated tracks are always added to the selection and prevent re-initialisations from the underlying detections through their interaction penalties.

5 Experimental Evaluation

We present experimental results on five different sequences recorded with three different platforms. In all cases, the sensor was a pair of forward-looking AVT Marlin F033C cameras, which deliver synchronised video streams of resolution 640 × 480 pixels at 13–14 frames per second. Bahnhofstrasse (999 frames) and Linthescher (1208 frames) have been recorded with a child stroller (baseline ≈ 0.4 m, sensor height ≈ 1 m, aperture angle ≈ 65°) in busy pedestrian zones, with people and street furniture frequently obstructing large portions of the field of view. Loewenplatz (800 frames), Bellevue (1500 frames), and City (3000 frames) have been recorded from a car (baseline ≈ 1 m, sensor height ≈ 1.3 m, aperture angle ≈ 50°) driving on inner-city streets among other cars, trucks and trams. Pedestrians appear mostly on sidewalks and crossings, and are observed only for short time spans. Lighting and contrast are realistic, with most sequences recorded on cloudy days in winter. Videos of tracking results are available as multimedia extensions, see appendix.

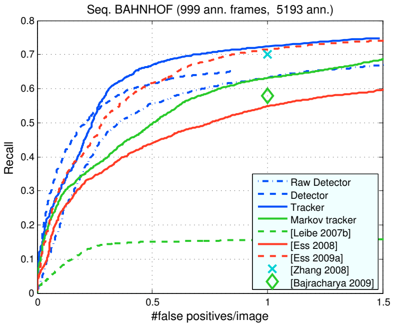


Fig. 6 Single-frame performance evaluation on Seq. Bahnhofstrasse. See text for details.

For testing, all system parameters are kept the same throughout all sequences, except for platform-specific parameters, such as the camera calibration and height and the ground plane prior (which depends on the wheelbase and suspension of the platform).

5.1 Quantitative Results.

Per-frame evaluation. In Fig. 6 we evaluate single-frame performance on Seq. Bahnhofstrasse, and compare the described method with its predecessors, as well as alternative approaches. In all cases, the bounding boxes estimated with different thresholds are compared to manually annotated ground truth by plotting recall over false positives per image. A bounding box is counted as correct if its intersection with the ground truth box is > 50% of their union.

The HOG detector alone, without any scene knowledge, already performs reasonably well ("raw detector"). Adding depth and ground-plane knowledge improves performance by 5–10% ("detector"). Adding tracking further improves the reachable recall, but loses performance in the high-precision regime (Ess et al, 2009a). This is partly an effect of per-frame evaluation: the tracker requires 2–3 detections to initialise a trajectory (losing recall), and it does report people while they are

[3] The exit zones are automatically shifted for a moving camera setup such that they always correspond to the image borders.

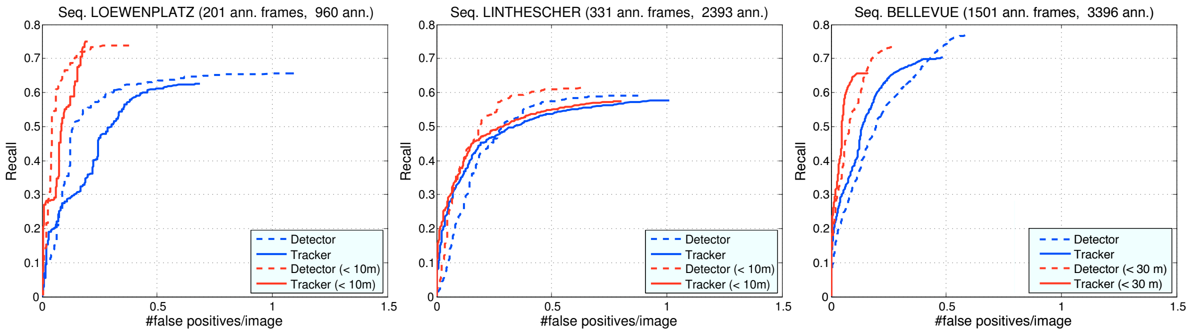


Fig. 7 Single-frame performance comparison between per-frame object detection (including 3D scene information), and object tracking, for four different sequences.

occluded and hence not annotated (losing precision). To compensate the latter effect, we also plot the detection rates after removing pedestrians who are occluded according to the estimated 3D state ("tracker"). The performance of the monocular system (Leibe et al, 2007b) using ISM and no depth information is very poor, mainly because of the high number of false alarms produced by the ISM pedestrian detector. Scene knowledge cleans up many of the false alarms, hence our previous ISM-based system with depth reasoning (Ess et al, 2008) reaches 55% at 1 false positive per image (FPPI). To assess the benefit of the multi-hypothesis approach, we also reduce our system to a first-order Markov tracker, by running only the extension step without any model selection (on the HOG detections), and initialising new trajectories from unassigned detections. This emulation, reaches 63% recall, similar to the raw detector, which shows the advantage of explicit multi-frame space-time reasoning.

On the same sequence Zhang et al (2008) report 70% recall at 1 FPPI. While they do not use stereo data, their approach is a batch process (requiring the detections of the entire video sequence) and can thus use future observations to correctly handle occlusions. Our online system performs comparably with 73% recall at 1 FPPI. Also using stereo data, Bajracharya et al (2009) report 58% recall on this sequence at 1 FPPI (and 42% recall on Seq. LINTHESCHER, see below).

In Fig. 7, we compare single-frame performance of the tracker with the scene-filtered detector (with stereo and ground-plane estimation, but without tracking) on three further sequences. Again, tracking suffers from the latency of trajectory initialisation (this effect is more pronounced for Seq. LOEWENPLATZ,which contains many briefly visible pedestrians). However, only the tracking stage can provide the necessary temporal information for motion prediction and dynamic path planning. The blue curves in Fig. 7 show the performance on all annotated pedestrians. When only considering the near and mid range up to 10/30 m distance (depending on the platform and driving speed), performance is considerably better, as indicated by the red curves.

	FPPI	full depth range		restricted to 15 m	
		Detector	Tracker	Detector	Tracker
no depth	0.5	—	0.19	—	0.32
	1.0	—	0.29	—	0.47
PS	0.5	0.63	0.60	0.66	0.66
	1.0	0.68	0.70	0.67	0.74
BP	0.5	0.65	0.64	0.66	0.73
	1.0	0.67	0.73	0.66	0.77
Zach	0.5	0.65	0.64	0.67	0.73
	1.0	0.67	0.73	0.67	0.78

Table 1 Detection rates for Seq. BAHNHOFSTRASSE with different stereo matching methods. Better depth maps improve localisation, and hence tracking, in the near field. Fast stereo methods come at the expense of slightly worse performance. Since we use robust statistics on depth, elaborate stereo algorithms bring little improvement at the system level.

In Table 1, we also compare the effect of using different stereo-matching methods for depth estimation (Sec. 3). This is of special interest, since nowadays a plethora of stereo algorithms of varying quality and runtime are available. Specifically, we compare fast GPU-based plane sweep stereo ("PS", Cornelis and Van Gool, 2005) with the widely used belief-propagation algorithm of Felzenszwalb and Huttenlocher. ("BP", 2006), and with a recent top-of-the-line method ("Zach", Zach et al, 2009), also using the GPU. On the one hand, modern algorithms indeed yield improvements in both scene analysis and tracking performance, but they come at the cost of considerably higher runtime (20 ms for PS vs. 30 s for Zach). On the other hand, depth maps are only an intermediate result in our pipeline and are processed with robust statistics. Therefore top-of-the-line stereo matching does not yield huge improvements in system performance, despite producing visibly better depth maps.

Track-level Evaluation. For a more fine-grained analysis, it is instructive to look at the tracking results at trajectory level. Automatic track-level evaluation of complex scenes is still an unsolved problem. We therefore resort to an interactive solution, which takes care of the obvious assignments between tracking output and ground-truth automatically, but which polls the user when ambiguities arise. The distance between two trajectories is defined as the distance between their corresponding object positions, robustly[4] averaged over the two trajectories' common lifetime. Formally,

$$d(H_j, H_k) = \frac{1}{|H_j \cap H_k|} \sum_{t_i \in H_j \cap H_k} \min\left(\|\mathbf{x}_j^{t_i} - \mathbf{x}_k^{t_i}\|, d_{max}\right) \ .$$

	BAHNHOFSTRASSE	LOEWENPLATZ
ground truth	89	107
tracker	125	126
mostly tracked	0.55	0.48
partially tracked	0.30	0.27
mostly missed	0.15	0.25
false alarms	0.62	1.09
ID switches	16	6
latency	9.9/1.5	0.3/2.0

Table 2 Trajectory-based evaluation on Seq. BAHNHOFSTRASSE and Seq. LOEWENPLATZ.

The metrics themselves then resemble the ones used in other trajectory-level evaluations (Wu and Nevatia, 2007; Li et al, 2009): as background information, Table 2 first reports the number of ground truth trajectories (GT) and the number of trajectories output by the tracker (OT). Then the fraction of mostly tracked (resp. missed) subjects is reported: each ground truth subject is classified as either mostly tracked (best output trajectory covers > 80% of the ground truth), partially tracked (output covers 20–80% of the ground truth), or mostly missed (estimate covers < 20% of the ground truth). Furthermore, we report the average number of false alarms *per frame*, the total number of identity switches (*i.e.* cases where a trajectory with a new label is started although the subject is still the same), as well as the latency (the mean and median number of frames until a trajectory is initialised after a subject enters the field of view).

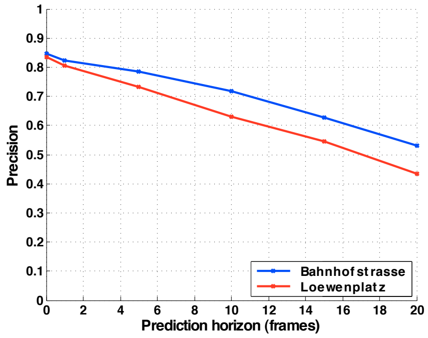


Fig. 8 Precision of the tracker prediction for increasing prediction horizon. Data was recorded at 12–14 fps.

In both cases, few false alarms occur. Also, the fraction of severe failures ("mostly missed") is relatively low. The fraction of only "partially tracked" subjects as well as the mean latency are high. This is due to the strict annotation: it often happens that a distant pedestrian is visible for a few frames, then becomes occluded for a long time before becoming visible at a much smaller distance, whence he is picked up by the tracker. Even in the best case, such a pedestrian will produce an identity switch, since the occlusion lasts too long to associate the two trajectories before and after it. In the worst case, however, the subject will only be picked up after leaving the occlusion, and hence be reported as "mostly missed" or "partially tracked". This is confirmed by the mean latency, which is significantly higher than the median, because the entire track of these subjects before being detected counts as latency. In Seq. BAHNHOFSTRASSE, 9 out of 76 (mostly or partially tracked) persons suffer from the aforementioned annotation problem and thus have latencies > 30 frames, severely biasing the mean latency. In fact, most of the "partially tracked" subjects are quite well covered: reducing the threshold for "mostly tracked" to 70% would increase the corresponding fraction to 0.72.

Prediction. To assess the suitability for dynamic path planning, we have also tested the accuracy of motion prediction from the estimated trajectories for increasing time horizons. This experiment is interesting, since it allows one to quantify the possible improvement compared to modelling all obstacles as static. We compare the bounding boxes predicted by the tracker with the actual annotations and count the fraction of false positives ($1 - precision$). The results are shown in Fig. 8. As expected, precision drops with increasing look-ahead, but stays within acceptable limits for a prediction horizon up to 1 second (12 frames). The experiment confirms that for reasonable prediction horizons, the accuracy does not drop greatly. This plot should be read qualitatively: a precision of 0.9 does not imply erroneous re-planning in every 10th frame, because many predicted objects do not affect the planned path.

[4] The gating at distance $d_{max} = 1\,\mathrm{m}$ is required to be robust against inaccuracies of the *ground truth*—objects are annotated in 2D, their depth has to be estimated from the bounding box.

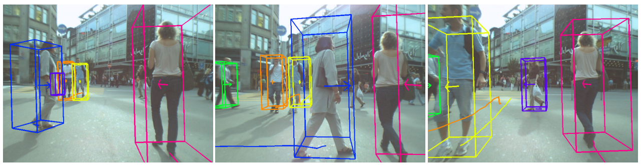
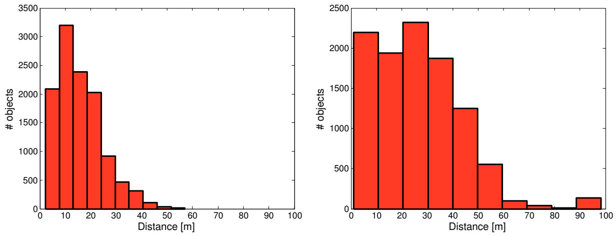


Fig. 9 Exemplary pedestrian tracking results on Seq. Linthescher.



(a) (b)

Fig. 13 Distribution of platform-to-object distances for pedestrians (a) and cars (b) on Seq. Bellevue.

5.2 Qualitative Results

Example results for Seq. Linthescher are shown in Fig. 9. This example highlights tracking through occlusions: the woman entering from the left temporarily occludes every other subject.

Example results for Seq. Bahnhofstrasse are shown in Fig. 10. 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 walks diagonally towards the camera (tracked with a pink bounding box). 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 our system—when no detections are available, its results simply revert to those of a depth-integration based occupancy map.

Fig. 11 demonstrates pedestrian tracking from a car. Compared to the previous sequences, the viewpoint is somewhat higher. Pedestrians either cross from left to right or are passed at high speed, therefore they stay in the field of view for fewer frames. Nevertheless the majority of pedestrians is tracked correctly.

Combined pedestrian and car tracking is demonstrated in Fig. 12 for Seq. City and Seq. Bellevue. Tracking cars is more complicated, because they are on average more distant and require an ensemble of 5 different viewpoint-specific detectors. In Seq. City, a further detector was trained for rear views of vans, in order to be able to track vans like the one in the top row of Fig. 12. Finally, Fig. 13 shows the depth distribution of tracked pedestrians and cars for Seq. Bellevue. Especially cars in many cases appear at distances of 30–50 m, where the localisation accuracy is as bad as 2–5 m in depth, making trajectory generation with smooth motion models difficult.

Computational Efficiency. For the real-time requirements of robotics or autonomous driving, computational efficiency is of prime importance. The current implementation of the tracking system does not fully reach this goal, in spite of fast GPU-implementations of stereo matching and visual odometry. The bottleneck is object detection with HOG (6 seconds per image). However, recently a parallel HOG-implementation on the GPU has been presented (Wojek et al, 2008). We have not yet integrated GPU-HOG into our software, but estimate the integration will bring down the processing time for a frame below 0.5 seconds. A real-time system in the near future thus seems within reach.

6 Concluding Remarks

We have described an object tracking system for dynamic path planning and mobile navigation. The system operates in world coordinates on a dynamically varying ground plane. The core tracking part of the system uses as observations the output of appearance-based detectors for the relevant object categories, in our case pedestrians and cars. At each frame, a redundant set of candidate trajectories is generated by starting standard EKF-trackers from different observations in the past and present. At this stage, the semantic information from object category detection allows our approach to employ the correct motion model for the object at hand. In a second stage, the set of trajectory candidates is pruned to a set with maximal joint likelihood using model selection. Two main characteristics of the proposed tracking system are that it continuously processes observations from a long time interval, rather than only considering the immediate past, and that it enables physically correct modelling of space-time violations, rather than only enforcing a unique data association. These extensions significantly improve tracking in the presence of realistic, complex interactions between multiple objects, including prolonged occlusion. The system has been evaluated on challenging sequences and manages to track most objects of interest over extended periods of time. The experiments support our claim that reliable tracking from a mobile observer in busy urban scenarios is possible and that robust systems are within reach.

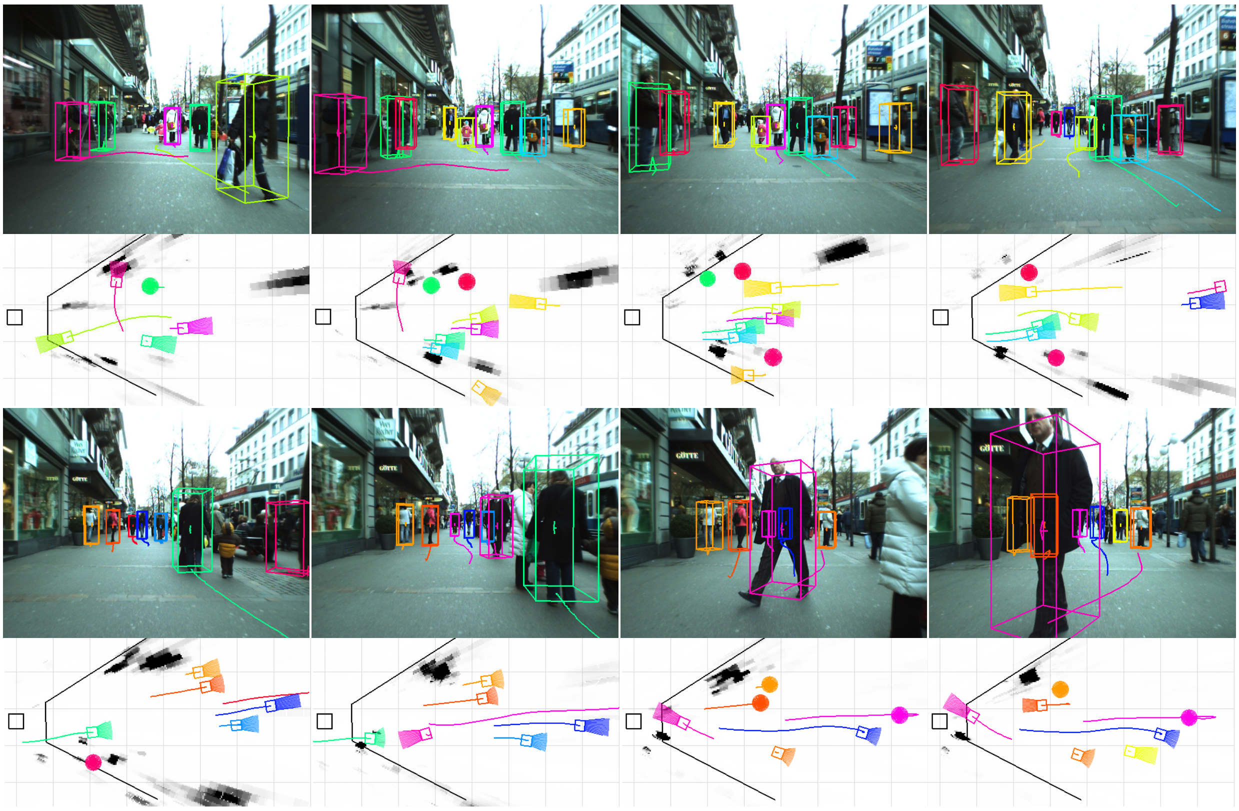


Fig. 10 Example results (pedestrians only) for Seq. Bahnhofstrasse.

Various extensions could potentially improve the current performance: the employed object detector proved to be surprisingly powerful, but object category detection is still an active field of research, and performance gains through more complete and more accurate detectors can be expected. In the long-term it would also make sense to focus only on the relevant scene agents, rather than track all of them, however this will require significant progress in modelling visual attention, which is still a problem.

Also, the employed motion models are rather simple and might be unable to handle erratic motions. More importantly, they do not take any information about the environment into account, although the path planning of real pedestrians and cars, respectively, is certainly influenced by the surrounding scene. Another obvious extension would be a discriminatively trained appearance model in order to better keep different scene objects apart. Last but not least, the success of model selection depends on a compact set of hypotheses, which nevertheless contain strong instances of all correct trajectories. Here, ideas from other global trackers could be implemented to, *e.g.*, link secure tracklets and prevent unnecessary hypothesis generation by applying information theoretic methods that spend more time sampling in difficult regions of the hypothesis space. Another challenge is the inclusion of occlusion reasoning directly into the model selection.

On the whole, we believe that visual tracking, together with other scene understanding capabilities which are also steadily improving, will play an important role for future autonomous driving and robotics.

Acknowledgements This project has been funded in parts by Toyota Motor Corporation and the EU projects DIRAC (IST-027787) and EUROPA (ICT-2008-231888). We thank Nico Cornelis and Christopher Zach for providing GPU implementations of their stereo matching algorithms. We also thank Kristijan Macek, Luciano Spinello, and Roland Siegwart for the opportunity to record from the SmartTer platform.

A Index to Multimedia Extensions

Multimedia extensions to this article are at `http://www.ijrr.org`.

1	video	tracking result on sequence Bahnhofstrasse
2	video	tracking result on sequence Linthescher
3	video	tracking result on sequence Loewenplatz

References

Ali S, Shah M (2008) Floor fields for tracking in high density crowd scenes. In: ECCV

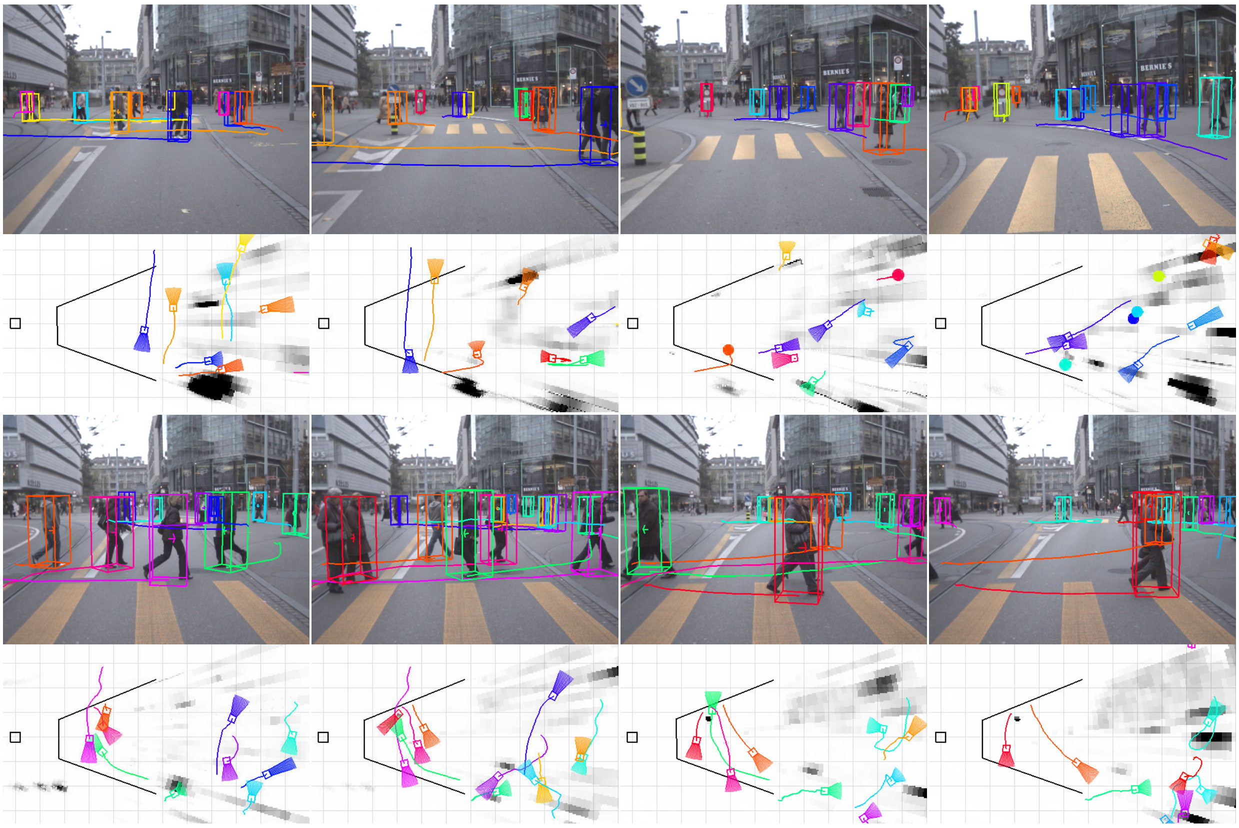


Fig. 11 Example results (pedestrians only, recorded from a car) for Seq. Loewenplatz.

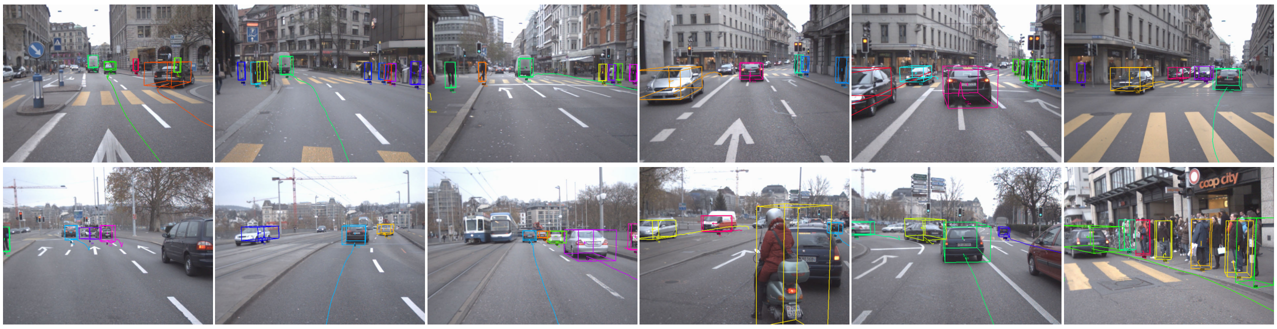


Fig. 12 Example results (pedestrians and cars) on Seq. City (top row) and Seq. Bellevue (bottom row).

Arras KO, Castellanos JA, Schilt M, Siegwart R (2003) Feature-based multi-hypothesis localization and tracking using geometric constraints. Robotics and Autonomous Systems 44

Arulampalam MS, Maskell S, Gordon N, Clapp T (2002) A tutorial on particle filters for online nonlinear/non-gaussian bayesian tracking. IEEE T Signal Proc 50:174–188

Avidan S (2005) Ensemble tracking. In: CVPR

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

Bajracharya M, Moghaddam B, Howard A, Brennan S, Matthies LH (2009) A fast stereo-based system for detecting and tracking pedestrians from a moving vehicle. Int JRobot Res 28:1466–1485

Berclaz J, Fleuret F, Fua P (2006) Robust people tracking with global trajectory optimization. In: CVPR

Betke M, Haritaoglu E, Davis LS (2000) Real-time multiple vehicle tracking from a moving vehicle. Mach Vision Appl 12(2):69–83

Bibby C, Reid I (2008) Robust real-time visual tracking using pixel-wise posteriors. In: ECCV

Boros E, Hammer PL (2002) Pseudo-boolean optimization. Discrete Appl Math 123(1-3):155–225

Breitenstein M, Reichlin F, Van Gool L (2009) Robust tracking-by-detection using a detector confidence particle filter. In: ICCV

Cameron S, Proberdt P (1994) Advanced Guided Vehicles, Aspects of the Oxford AGV Project. World Scientific, Singapore

Comaniciu D, Ramesh V, Meer P (2003) Kernel-based object tracking. IEEE T Pattern Anal 25(5):564–575

Cornelis N, Van Gool L (2005) Real-time connectivity constrained depth map computation using programmable graphics hardware. In: CVPR

Cox IJ (1993) A review of statistical data association techniques for motion correspondence. Int J Comput Vision 10(1):53–66

Dalal N, Triggs B (2005) Histograms of oriented gradients for human detection. In: CVPR

DARPA (2008) DARPA urban challenge rulebook. http://www.darpa.mil/GRANDCHALLENGE/

Dollar P, Wojek C, Schiele B, Perona P (2009) Pedestrian detection: A benchmark. In: CVPR

Enzweiler M, Gavrila DM (2009) Monocular pedestrian detection: Survey and experiments. IEEE T Pattern Anal

Ess A, Leibe B, Van Gool L (2007) Depth and appearance for mobile scene analysis. In: ICCV

Ess A, Leibe B, Schindler K, Van Gool L (2008) A mobile vision system for robust multi-person tracking. In: CVPR

Ess A, Leibe B, Schindler K, Van Gool L (2009a) Moving obstacle detection in highly dynamic scenes. In: ICRA

Ess A, Leibe B, Schindler K, Van Gool L (2009b) Robust multi-person tracking from a mobile platform. IEEE T Pattern Anal 31(10):1831–1846

Ess A, Schindler K, Leibe B, Van Gool L (2009c) Improved multi-person tracking with active occlusion handling. In: ICRA Workshop on People Detection and Tracking

Everingham M, Van Gool L, Williams CKI, Winn J, Zisserman A (2008) The PASCAL Visual Object Classes Challenge 2008 (VOC2008) Results. http://www.pascal-network.org/challenges/VOC/voc2008/workshop/index.html

Felzenszwalb P, McAllester D, Ramanan D (2008) A discriminatively trained, multiscale, deformable part model. In: CVPR

Felzenszwalb PF, Huttenlocher DP (2006) Efficient belief propagation for early vision. Int J Comput Vision 70:41–54, available from http://people.cs.uchicago.edu/ pff/bp/

Fortmann TE, Bar Shalom Y, Scheffe M (1983) Sonar tracking of multiple targets using joint probabilistic data association. IEEE J Ocean Eng 8(3):173–184

Gavrila DM, Munder S (2007) Multi-cue pedestrian detection and tracking from a moving vehicle. Int J Comput Vision 73:41–59

Gelb A (1996) Applied Optimal Estimation. MIT Press

Grabner H, Bischof H (2006) On-line boosting and vision. In: CVPR

Isard M, Blake A (1998) CONDENSATION–conditional density propagation for visual tracking. In: Int J Comput Vision, vol 29(1)

Isard M, MacCormick J (2001) Bramble: A bayesian multiple-blob tracker. In: ICCV

Jiang H, Fels S, Little JJ (2007) A linear programming approach for multiple object tracking. In: CVPR

Kaucic R, Perera AG, Brooksby G, Kaufhold J, Hoogs A (2005) A unified framework for tracking through occlusions and across sensor gaps. In: CVPR

Khan Z, Balch T, Dellaert F (2005) Mcmc-based particle filtering for tracking a variable number of interacting targets. IEEE T Pattern Anal 27(11):1805–1819

Lanz O (2006) Approximate bayesian multibody tracking. IEEE T Pattern Anal 28(9):1436–1449

Leibe B, Cornelis N, Cornelis K, Van Gool L (2007a) Dynamic 3d scene analysis from a moving vehicle. In: CVPR

Leibe B, Schindler K, Van Gool L (2007b) Coupled detection and trajectory estimation for multi-object tracking. In: ICCV

Leibe B, Leonardis A, Schiele B (2008a) Robust object detection with interleaved categorization and segmentation. Int J Comput Vision 77(1-3):259–289

Leibe B, Schindler K, Cornelis N, Van Gool L (2008b) Coupled detection and tracking from static cameras and moving vehicles. IEEE T Pattern Anal 30(10):1683–1698

Li Y, Huang C, Nevatia R (2009) Learning to associate: HybridBoosted multi-target tracker for crowded scene. In: CVPR

Morefield CL (1977) Application of 0-1 integer programming to multitarget tracking problems. IEEE T Automat Contr

Munkres J (1957) Algorithms for the assignment and transportation problems. J SIAM 5:32–38

Murty KG (1968) An algorithm for ranking all the assignments in order of increasing cost. Oper Res 16:682–687

Nillius P, Sullivan J, Carlsson S (2006) Multi-target tracking-linking identities using Bayesian network inference. In: CVPR

Nistér D, Naroditsky O, Bergen JR (2004) Visual odometry. In: CVPR

Nummiaro K, Koller-Meier E, Gool LV (2003) An adaptive color-based particle filter. Image Vision Comput 21(1):99–110

Okuma K, Taleghani A, de Freitas N, Little J, Lowe D (2004) A boosted particle filter: Multitarget detection and tracking. In: ECCV

Perera AGA, Srinivas C, Hoogs A, Brooksby G, Hu W (2006) Multi-object tracking through simultaneous long occlusions and split-merge conditions. In: CVPR

Reid DB (1979) An algorithm for tracking multiple targets. IEEE T Automat Contr 24(6):843–854

Rother C, Kolmogorov V, Lempitsky VS, Szummer M (2007) Optimizing binary mrfs via extended roof duality. In: CVPR

Ryoo MS, Aggarwal JK (2008) Observe-and-explain: A new approach for multiple hypotheses tracking of humans and objects. In: CVPR

Schindler K, U J, Wang H (2006) Perspective n-view multibody structure-and-motion through model selection. In: ECCV

Schrijver A (1998) Theory of Linear and Integer Programming. John Wiley & Sons

Schulz D, Burgard W, Fox D, Cremers AB (2001) Tracking multiple moving targets with a mobile robot using particle filters and statistical data association. In: ICRA

Song X, Cui J, Zha H, Zhao H (2008) Vision-based multiple interacting targets tracking via on-line supervised learning. In: ECCV

Stauffer C, Grimson WEL (1999) Adaptive background mixture models for real-time tracking. In: CVPR

Toyama K, Krumm J, Brumitt B, Meyers B (1999) Wallflower: principles and practice of background maintenance. In: ICCV

Viterbi A (1967) Error bounds for convolutional codes and an asymptotically optimum decoding algorithm. IEEE T Inform Theory 13(2):260–269

Wojek C, Schiele B (2008) A dynamic CRF model for joint labeling of object and scene classes. In: ECCV

Wojek C, Dorkó G, Schulz A, Schiele B (2008) Sliding-windows for rapid object class localization: A parallel technique. In: DAGM

Wolf JK, Viterbi AM, Dixson GS (1989) Finding the best set of K paths through a trellis with application to multitarget tracking. IEEE T Aero Elec Sys 25(29):287–295

Wu B, Nevatia R (2007) Detection and tracking of multiple, partially occluded humans by bayesian combination of edgelet part detectors. Int J Comput Vision 75(2):247–266

Yan F, Kostin A, Christmas WJ, Kittler J (2006) A novel data association algorithm for object tracking in clutter with application to tennis video analysis. In: CVPR

Yu Q, Medioni G, Cohen I (2007) Multiple target tracking using spatio-temporal Markov chain Monte Carlo data association. In: CVPR

Zach C, Frahm JM, Niethammer M (2009) Continuous maximal flows and Wulff shapes: Application to MRFs. In: CVPR

Zhang L, Li Y, Nevatia R (2008) Global data association for multi-object tracking using network flows. In: CVPR

Zhao T, Nevatia R, Wu B (2008) Segmentation and tracking of multiple humans in crowded environments. IEEE T Pattern Anal 30(7):1198–1211