Coupled Object Detection and Tracking from Static Cameras and Moving Vehicles

Bastian Leibe, Konrad Schindler, Nico Cornelis, and Luc Van Gool

***Abstract*— We present a novel approach for multi-object tracking which considers object detection and spacetime trajectory estimation as a coupled optimization problem. Our approach is formulated in a Minimum Description Length hypothesis selection framework, which allows our system to recover from mismatches and temporarily lost tracks. Building upon a state-of-the-art object detector, it performs multi-view/multi-category object recognition to detect cars and pedestrians in the input images. The 2D object detections are checked for their consistency with (automatically estimated) scene geometry and are converted to 3D observations, which are accumulated in a world coordinate frame. A subsequent trajectory estimation module analyzes the resulting 3D observations to find physically plausible spacetime trajectories. Tracking is achieved by performing model selection after every frame. At each time instant, our approach searches for the globally optimal set of spacetime trajectories which provides the best explanation for the current image and for all evidence collected so far, while satisfying the constraints that no two objects may occupy the same physical space, nor explain the same image pixels at any point in time. Successful trajectory hypotheses are then fed back to guide object detection in future frames. The optimization procedure is kept efficient through incremental computation and conservative hypothesis pruning. We evaluate our approach on several challenging video sequences and demonstrate its performance on both a surveillance-type scenario and a scenario where the input videos are taken from inside a moving vehicle passing through crowded city areas.**

***Index Terms*— Object Detection, Tracking, Model Selection, MDL, Structure-from-Motion, Mobile Vision**

I. Introduction

Multi-object tracking is a challenging problem with numerous important applications. The task is to estimate multiple interacting object trajectories from video input, either in the 2D image plane or in 3D object space. Typically, tracking is modeled as some kind of first-order Markov chain, *i.e.* object locations at a time step t are predicted from those at the previous time step $(t{-}1)$ and then refined by comparing the object models to the current image data, whereupon the object models are updated and the procedure is repeated for the next time step. The Markov paradigm implies that trackers cannot recover from failure, since once they have lost track, the information handed on to the next time step is wrong. This is a particular problem in a multi-object scenario, where object-object interactions and occlusions are likely to occur.

Several approaches have been proposed to work around this restriction. Classic multi-target trackers such as Multi-Hypothesis Tracking (MHT) [39] and Joint Probabilistic Data Association Filters (JPDAFs) [13] jointly consider the data association from sensor measurements to multiple overlapping tracks. While not restricted to first-order Markov chains, they can however only keep few time steps in memory due to the exponential task complexity. Moreover, originally developed for point targets, those approaches generally do not take physical exclusion constraints between object volumes into account.

The other main difficulty is to identify which image parts correspond to the objects to be tracked. In classic surveillance settings with static cameras, this task can often be addressed by background modelling (*e.g.* [42]). However, this is no longer the case when large-scale background changes are likely to occur, or when the camera itself is moving. In order to deal with such cases and avoid drift, it becomes necessary to combine tracking with detection.

This has only recently become feasible due to the rapid progress of object (class) detection [10], [31], [43], [46]. The idea behind such a combination is to run an object detector, trained either offline to detect an entire object category or online to detect specific objects [1], [17]. Its output can then constrain the trajectory search to promising image regions and serve to re-initialize in case of failure. Going one step further, one can directly use the detector output as data source for tracking (instead of *e.g.* color information).

In this paper, we will specifically address multi-object tracking both from static cameras and from a moving, camera-equipped vehicle. Scene analysis of this sort requires multi-viewpoint, multi-category object detection. Since we cannot control the vehicle's path, nor the environment it passes through, the detectors need to be robust to a large range of lighting variations, noise, clutter, and partial occlusion. In order to localize the detected objects in 3D, an accurate estimate of the scene geometry is necessary. The ability to integrate such measurements over time additionally requires continuous self-localization and recalibration. In order to finally make predictions about future states, powerful tracking is needed that can cope with a changing background.

We address those challenges by integrating recognition, reconstruction, and tracking in a collaborative ensemble. Namely, we use Structure-from-Motion (SfM) to estimate scene geometry at each time step, which greatly helps the other modules. Recognition picks out objects of interest and separates them from the dynamically changing background. Tracking adds a temporal context to individual object detections and provides them with a history supporting their presence in the current video frame. Detected object trajectories, finally, are extrapolated to future frames in order to guide detection there.

In order to improve robustness, we further propose to couple object detection and tracking in a non-Markovian hypothesis selection framework. Our approach implements a feedback loop, which passes on predicted object locations as a prior to influence detection in future frames, while at the same time choosing between and reevaluating trajectory hypotheses in the light of new evidence. In contrast to previous approaches, which optimize individual trajectories in a temporal window [2], [47] or over

B. Leibe, K. Schindler, and L. Van Gool are with the Computer Vision Laboratory at ETH Zurich.

N. Cornelis and L. Van Gool are with ESAT/PSI-IBBT at KU Leuven.

sensor gaps [22], our approach tries to find a globally optimal combined solution for all detections and trajectories, while incorporating real-world physical constraints such that no two objects can occupy the same physical space, nor explain the same image pixels at the same time. The task complexity is reduced by only selecting between a limited set of plausible hypotheses, which makes the approach computationally feasible.

The paper is structured as follows. After discussing related work in Section II, Section III describes the Structure-from-Motion system we use for estimating scene geometry. Section IV then presents our hypothesis selection framework integrating object detection and trajectory estimation. Sections V and VI introduce the baseline systems we employ for each of those components, after which Section VII presents our coupled formulation as a combined optimization problem. Several important implementation details are discussed in Section VIII. Section IX finally presents experimental results.

II. Related Work.

In this paper, we address multi-object tracking in two scenarios. First, we will demonstrate our approach in a typical surveillance scenario with a single, static, calibrated camera. Next, we will apply our method to the challenging task of detecting, localizing, and tracking other traffic participants from a moving vehicle.

Tracking in such scenarios consists of two subproblems: trajectory initialization and target following. While many approaches rely on background subtraction from a static camera for the former (*e.g.* [2], [26], [42]), several recent approaches have started to explore the possibilities of combining tracking with detection [1], [17], [37], [46]. This has been helped by the considerable progress of object detection over the last few years [10], [31], [34], [43]–[45], which has resulted in state-of-the-art detectors that are applicable in complex outdoor scenes.

The second subproblem is typically addressed by classic tracking approaches, such as Extended Kalman Filters (EKF) [15], particle filtering [21], or Mean-Shift tracking [6], which rely on a Markov assumption and carry the associated danger of drifting away from the correct target. This danger can be reduced by optimizing data assignment and considering information over several time steps, as in MHT [9], [39] and JPDAF [13]. However, their combinatorial nature limits those approaches to consider either only few time steps [39] or only single trajectories over longer time windows [2], [22], [47]. In contrast, our approach simultaneously optimizes detection and trajectory estimation for multiple interacting objects and over long time windows by operating in a hypothesis selection framework.

Tracking with a moving camera is a notoriously difficult task because of the combined effects of egomotion, blur, and rapidly changing lighting conditions [3], [14]. In addition, the introduction of a moving camera invalidates many simplifying techniques we have grown fond of, such as background subtraction and a constant ground plane assumption. Such techniques have been routinely used in surveillance and tracking applications from static cameras (*e.g.* [2], [24]), but they are no longer applicable here. While object tracking under such conditions has been demonstrated in clean highway situations [3], reliable performance in urban areas is still an open challenge [16], [38].

Clearly, every source of information that can help system performance under those circumstances constitutes a valuable aid that should be used. Hoiem *et al*. [20] have shown that scene geometry can fill this role and greatly help recognition. They describe a method how geometric scene context can be automatically estimated from a single image [19] and how it can be used for improving object detection performance. More recently, Cornelis *et al*. have shown how recognition can be combined with Structure-from-Motion for the purpose of localizing static objects [8]. In this paper, we extend the framework developed there in order to also track moving objects.

Our approach integrates geometry estimation and tracking-by-detection in a combined system that searches for the best global scene interpretation by joint optimization. Berclaz *et al*. [2] also perform trajectory optimization to track up to six mutually occluding individuals by modelling their positions on a discrete occupancy grid. However, their approach requires multiple static cameras, and optimization is performed only for one individual at a time. In contrast, our approach models object positions continuously while moving through a 3D world and allows to find a jointly optimal solution.

III. Online Scene Geometry Estimation

Our combined tracking-by-detection approach makes the following uses of automatically estimated scene geometry information. First, it employs the knowledge about the scene's ground plane in order to restrict possible object locations during detection. Second, a camera calibration allows us to integrate individual detections over time in a world coordinate frame and group them into trajectories over a spacetime window. Such a calibration can be safely assumed to be available when working with a static camera, as in typical surveillance scenarios. However, this is no longer the case when the camera itself is moving. In this paper, we show that 3D tracking is still possible from a moving vehicle through a close combination with Structure-from-Motion (SfM). Taking as input two video streams from a calibrated stereo rig mounted on the vehicle's roof, our approach uses SfM in order to continually estimate the camera pose and ground plane for every frame. This is done as follows.

A. Real-Time Structure-from-Motion (SfM).

Our SfM module is based on the approach by [7], [8], which is highly optimized and runs at ≈ 30 frames per second. Feature points are extracted with a simple, but extremely fast interest point operator, which divides local neighborhoods into four subtiles and compares their average intensities.The extracted features are matched between consecutive images and then fed into a classic SfM pipeline [18], which reconstructs feature tracks and refines 3D point locations by triangulation. A windowed bundle adjustment is running in parallel with the main SfM algorithm to refine camera poses and 3D feature locations for previous frames and thus reduce drift.

B. Online Ground Plane Estimation.

For each image pair, SfM delivers an updated camera calibration. In addition, we obtain an online estimate of the ground plane by fitting trapezoidal patches to the reconstructed wheel contact points of adjacent frames, and by smoothing their normals over a larger spatial window (see Fig. 1). Empirically, averaging the normals over a length of 3m (or roughly the wheel-base of the vehicle) turned out to be optimal for a variety of cases. Note that using a constant *spatial* window automatically adjusts for

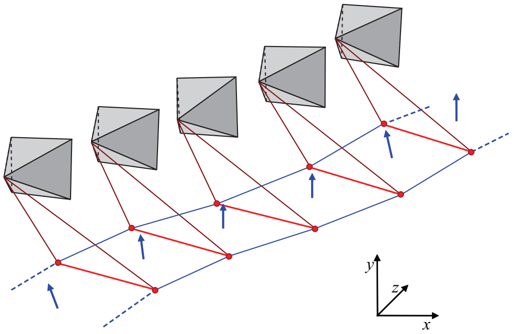


Fig. 1. *Visualization of the online ground plane estimation procedure. Using the camera positions from SfM, we reconstruct trapezoidal road strips between the car's wheel contact points of adjacent frames. A ground plane estimate is obtained by averaging the local normals over a spatial window of about 3m travel distance.*

driving speed: reconstructions are more accurate at low speed, respectively high frame rate (once the 3D structure has stabilized after initialization), so that smaller road patches are sufficient for estimating the normal.

Figure 2 highlights the importance of this continuous reestimation step if later stages are to trust its results. In this example, the camera vehicle hits a speedbump, causing a massive jolt in camera perspective. The top row of Fig. 2 shows the resulting detections when the ground plane estimate from the previous frame is simply kept fixed. As can be seen, this results in several false positives at improbable locations and scales. The bottom image displays the detections when the reestimated ground plane is used instead. Here, the negative effect is considerably lessened.

IV. Approach

A. MDL Hypothesis Selection.

Our basic mathematical tool is a model selection framework as introduced in [32] and adapted in [28]. We briefly repeat its general form here and later explain specific versions for object detection and trajectory estimation.

The intuition of the method is that in order to correctly handle the interactions between multiple models required to describe a data set, one cannot fit them sequentially (because interactions with models which have not yet been estimated would be neglected). Instead, an over-complete set of hypothetical models is generated, and the best subset is chosen with model selection in the spirit of the minimum description length (MDL) criterion.

To select the best models, the *savings* (in coding length) of each hypothesis h are expressed as

$$S_h \sim S_{data} - \kappa_1 S_{model} - \kappa_2 S_{error} \ , \qquad (1)$$

where S_{data} corresponds to the number N of data points, which are explained by h; S_{model} denotes the cost of coding the model itself; S_{error} describes the cost for the error committed by the representation; and κ_1, κ_2 are constants to weigh the different factors. If the error term is chosen as the log-likelihood over all data points x assigned to a hypothesis h, then the following approximation holds[1]:

$$S_{error} = -\log \prod_{x\in h} p(x|h) = -\sum_{x\in h} \log p(x|h) \qquad (2)$$

$$= \sum_{x\in h} \sum_{n=1}^{\infty} \frac{1}{n}(1-p(x|h))^n \approx N - \sum_{x\in h} p(x|h).$$

Substituting eq.(2) into eq.(1) yields an expression for the merit of model h:

$$S_h \sim -\kappa_1 S_{model} + \sum_{x\in h} \left((1-\kappa_2) + \kappa_2 p(x|h)\right) \ . \qquad (3)$$

Thus, the merit of a putative model is essentially the sum over its data assignment likelihoods, regularized with a term which compensates for unequal sampling of the data.

A data point can only be assigned to one model. Hence, overlapping hypothetical models compete for data points. This competition translates to interaction costs, which apply only if both hypotheses are selected and which are then subtracted from the score of the hypothesis combination. Leonardis *et al*. [32] have shown that if only pairwise interactions are considered[2], then the optimal set of models can be found by solving the Quadratic Boolean Problem (QBP)

$$\max_n n^\mathsf{T} S n \quad , \quad S = \begin{bmatrix} s_{11} & \cdots & s_{1N} \\ \vdots & \ddots & \vdots \\ s_{N_1} & \cdots & s_{NN} \end{bmatrix} . \qquad (4)$$

Here, $n = [n_1, n_2, \ldots, n_N]^\mathsf{T}$ is a vector of indicator variables, such that $n_i = 1$ if hypothesis h_i is accepted, and $n_i = 0$ otherwise. S is an interaction matrix, whose diagonal elements s_{ii} are the merit terms (3) of individual hypotheses, while the off-diagonal elements $(s_{ij} + s_{ji})$ express the interaction costs between two hypotheses h_i and h_j.

B. Object Detection.

For object detection, we use the *Implicit Shape Model* (ISM) detector of [28], [31], which utilizes the model selection framework explained above. It uses a voting scheme based on multi-scale interest points to generate a large number of hypothetical detections. From this redundant set, the subset with the highest joint likelihood is selected by maximizing $n^\mathsf{T} S n$: the binary vector n indicates which detection hypotheses shall be used to explain the image observations and which ones can be discarded. The interaction matrix S contains the hypotheses' individual

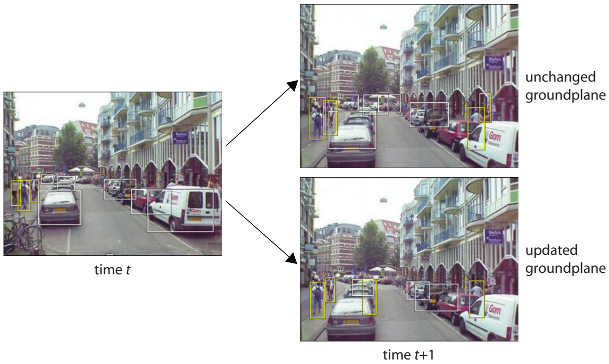


Fig. 2. *Illustration for the importance of a continuous reestimation of scene geometry. The images show the effect on object detection when the vehicle hits a speedbump (top) if using an unchanged ground plane estimate; (bottom) if using the online reestimate.*

[1] This approximation improves robustness against outliers by mitigating the non-linearity of the logarithm near 0, while providing good results for unambiguous point assignments.

[2] Considering only interactions between pairs of hypotheses is a good approximation, because their cost dominates the total interaction cost. Furthermore, neglecting higher order interactions always increases interaction costs, yielding a desirable bias against hypotheses with very little evidence.

savings, as well as their interaction costs, which encode the constraint that each image pixel is counted only as part of at most one detection. This module is described in detail in Section V.

C. Trajectory estimation.

In [27], a similar formalism is also applied to estimate object trajectories over the ground plane. Object detections in a 3D spacetime volume are linked to hypothetical trajectories with a simple dynamic model, and the best set of trajectories is selected from those hypotheses by solving another maximization problem $m^{\mathsf{T}}Qm$, where the interaction matrix Q again contains the individual savings and the interaction costs which arise if two hypotheses compete to fill the same part of the spacetime volume (see Section VI).

D. Coupled Detection and Trajectory estimation.

Thus, both object detection and trajectory estimation can be formulated as individual QBPs. However, as shown in [30], the two tasks are closely coupled, and their results can mutually reinforce each other. In Section VII, we therefore propose a combined formulation that integrates both components into a coupled optimization problem. This joint optimization searches for the best explanation of the current image and all previous observations, while allowing bidirectional interactions between those two parts. As our experiments in Section IX will show, the resulting feedback from tracking to detection improves total system performance and yields more stable tracks.

V. Object Detection

The recognition system is based on a battery of single-view, single-category ISM detectors [31]. This approach lets local features, extracted around interest regions, vote for the object center in a 3-dimensional Hough space, followed by a top-down segmentation and verification step. For our application, we use the robust multi-cue extension from [29], which integrates multiple local cues, in our case local *Shape Context* descriptors [35] computed at *Harris-Laplace*, *Hessian-Laplace*, and *DoG* interest regions [33], [35].

In order to capture different viewpoints of cars, our system uses a set of 5 single-view detectors trained for the viewpoints shown in Fig. 3(top) (for training efficiency we run mirrored versions of the two semi-profile detectors for the symmetric viewpoints). In addition, we use a pedestrian detector trained on both frontal and side views of pedestrians. The detection module does not differentiate between pedestrians and bicyclists here, as those two categories are often indistinguishable from a distance and our detector responds well to both of them. We start by running all detectors on both camera images and collect their hypotheses.For each such hypothesis h, we compute two per-pixel probability maps $p(\mathbf{p} = \mathit{figure}|h)$ and $p(\mathbf{p} = \mathit{ground}|h)$, as described in [31]. The rest of this section describes how the different detector outputs are fused and how scene geometry is integrated into the recognition system.

A. Integration of Scene Geometry Constraints

The integration with scene geometry follows the framework described in [8], [27]. With the help of a camera calibration, the 2D detections h are converted to 3D object locations H on the ground plane. This allows us to evaluate each hypothesis under a 3D location prior $p(H)$. The location prior is split up into a uniform distance prior for the detector's target range and a Gaussian prior for typical pedestrian sizes $p(H_{size}) \sim \mathcal{N}(1.7, 0.2^2)$ [meters], similar to [20].

This effective coupling between object distance and size through the use of a ground plane has several beneficial effects. First, it significantly reduces the search volume during voting to a corridor in Hough space (Fig. 3(bottom left)). In addition, the Gaussian size prior serves to "pull" object hypotheses towards the correct locations, thus improving also recognition quality.

B. Multi-Detector Integration

In contrast to [8], we fuse the outputs of the different single-view detectors already at this stage. This is done by expressing the per-pixel support probabilities $p(\mathbf{p}{=}\mathit{fig.}|H)$ by a marginalization over all image-plane hypotheses that are consistent with the same 3D object H.

$$p(\mathbf{p}{=}\mathit{fig.}|H) = \sum_j p(\mathbf{p}{=}\mathit{fig.}|h_j)p(h_j|H). \tag{5}$$

The new factor $p(h_j|H)$ is a 2D/3D transfer function, which relates the image-plane hypotheses h_j to the 3D object hypothesis H. We implement this factor by modeling the object location and main orientation of H with an oriented 3D Gaussian, as shown in Fig. 3(bottom right). Thus, multiple single-view detections can contribute to the same 3D object if they refer to a similar 3D location and orientation.

This step effectively makes use of symmetries in the different single-view detectors in order to increase overall system robustness. For example, the frontal and rear-view car detectors often respond to the same image structures because of symmetries in the car views. Similarly, a slightly oblique car view may lead to responses from both the frontal and a semi-profile detector. Rather than to have those hypotheses compete, our system lets them reinforce each other as long as they lead to the same interpretation of the underlying scene.

Finally, we express the score of each hypothesis in terms of the pixels it occupies. Let I be the image and $Seg(H)$ be the support region of H, as defined by the fused detections (*i.e.* the pixels for which $p(\mathbf{p} = \mathit{figure}|H) > p(\mathbf{p} = \mathit{ground}|H)$). Then

$$\begin{aligned} p(H|I) &\sim p(I|H)p(H) \\ &= p(H)\prod_{\mathbf{p}\in I} p(\mathbf{p}|H) = p(H) \prod_{\mathbf{p}\in Seg(H)} p(\mathbf{p}{=}\mathit{fig.}|H). \end{aligned} \tag{6}$$

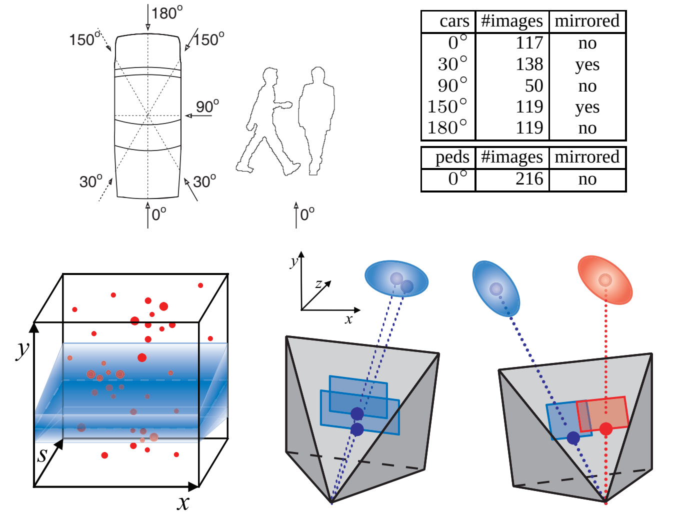
The updated hypotheses are then passed on to the following hypothesis selection stage.

C. Multi-Category Hypothesis Selection.

In order to obtain the final interpretation for the current image pair, we search for the combination of hypotheses that together best explain the observed evidence. This is done by adopting the MDL formulation from eq. (1), similar to [28], [31]. In contrast to that previous work, however, we perform the hypothesis selection not over image-plane hypotheses h_i, but over their corresponding world hypotheses H_i.

For notational convenience, we define the pseudo-likelihood

$$p^{*}(H|I) = \frac{1}{A_{s,v}} \sum_{\mathbf{p}\in Seg(H)} ((1-\kappa_2) + \kappa_2 p(\mathbf{p}{=}\mathit{fig.}|H)) + \log p(H)\ , \tag{7}$$



cars	#images	mirrored
0°	117	no
30°	138	yes
90°	50	no
150°	119	yes
180°	119	no

peds	#images	mirrored
0°	216	no

Fig. 3. *(top) Training viewpoints used for cars and pedestrians. (bottom left) The estimated ground plane significantly reduces the search volume for object detection. A Gaussian size prior additionally "pulls" object hypotheses towards the right locations. (bottom right) The responses of multiple detectors are combined if they refer to the same scene object*

where $A_{s,v}$ acts as a normalization factor expressing the *expected area* of an object hypothesis at its detected scale and aspect. The term $p(\mathbf{p}{=}\mathit{fig.}|H)$ integrates all consistent single-view detections, as described in the previous section.

Two detections H_i and H_j interact if they overlap and compete for the same image pixels. In this case, we assume that the hypothesis $H_k \in \{H_i, H_j\}$ that is farther away from the camera is occluded. Thus, the cost term subtracts H_k's support in the overlapping image area, thereby ensuring that only this area's contribution to the front hypothesis survives. With the approximation from eq. (2), we thus obtain the following merit and interaction terms for the object detection matrix S:

$$s_{ii} = -\kappa_1 + p^*(H_i|I) \qquad (8)$$

$$s_{ij} = -\frac{1}{2A_{s,v}} \sum_{\mathbf{p}\in Seg(H_i\cap H_j)} ((1-\kappa_2) + \kappa_2 p(\mathbf{p}{=}\mathit{fig.}|H_k)) - \frac{1}{2}\log p(H_k).$$

As a result of this procedure, we obtain a set of world hypotheses $\{H_i\}$, together with their supporting segmentations in the image. At the same time, the hypothesis selection procedure naturally integrates the contributions from the different single-view, single-category detectors.

VI. Spacetime Trajectory Estimation

In order to present our trajectory estimation approach, we introduce the concept of *event cones*. The event cone of an observation $H_{i,t} = \{\mathbf{x}_{i,t}, v_{i,t}, \theta_{i,t}\}$ is the spacetime volume it can physically influence from its current position given its maximal velocity and turn rate. Figure 4 shows an illustration for several cases of this concept. If an object is static at time t and its orientation is unknown, all motion directions are equally probable, and the affected spacetime volume is a simple double cone reaching both forwards and backwards in time (Fig. 4(a)). If the object moves holonomically, *i.e.* without external constraints linking its speed and turn rate, the event cone becomes tilted in the motion direction (Fig. 4(b)). An example for this case would be a pedestrian at low speeds. In the case of nonholonomic motion, as in a car which can only move along its main axis and only turn while moving, the event cones get additionally deformed according to those (often nonlinear) constraints (Fig. 4(c)).

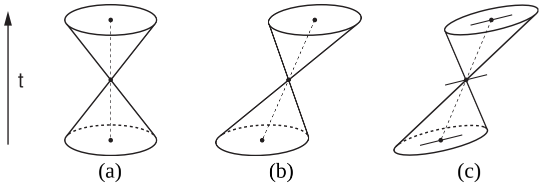


Fig. 4. *Visualization of example event cones for (a) a static object with unknown orientation; (b) a holonomically moving object; (c) a non-holonomically moving object.*

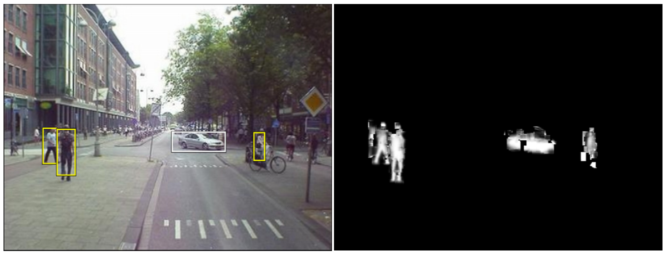


Fig. 5. *Detections and corresponding top-down segmentations used to learn the object-specific color model.*

We thus search for plausible trajectories through the spacetime observation volume by linking up event cones, as shown in Fig. 6. Starting from an observation $H_{i,t}$, we follow its event cone up and down the timeline and collect all observations that fall inside this volume in the adjoining time steps. Since we do not know the starting velocity $v_{i,t}$ yet, we begin with the case in Fig. 4(a). In all subsequent time steps, however, we can reestimate the object state from the new evidence and adapt the growing trajectory accordingly.

It is important to point out that an individual event cone is not more powerful in its descriptive abilities than a bidirectional Extended Kalman Filter, since it is based on essentially the same equations. However, our approach goes beyond Kalman Filters in several important respects. First of all, we are no longer bound by a Markov assumption. When reestimating the object state, we can take several previous time steps into account. In our approach, we aggregate the information from all previous time steps, weighted with a temporal discount λ. In addition, we are not restricted to tracking a single hypothesis. Instead, we start independent trajectory searches from all available observations (at all time steps) and collect the corresponding hypotheses. The final scene interpretation is then obtained by a global optimization stage which selects the combination of trajectory hypotheses that best explains the observed data under the constraints that each observation may only belong to a single object and no two objects may occupy the same physical space at the same time. The following sections explain those steps in more detail.

A. Color Model.

For each observation, we compute an object-specific color model a_i, using the top-down segmentations provided by the previous stage. Figure 5 shows an example of this input. For each detection $H_{i,t}$, we build an $8 \times 8 \times 8$ RGB color histogram a_i over the segmentation area, weighted by the per-pixel confidence $\sum_k p(\mathbf{p} = \mathit{fig.}|h_k)p(h_k|H_{i,t})$ in this segmentation. The appearance model $\mathcal{A}$ is defined as the trajectory's color histogram. It is initialized with the first detection's color histogram and then evolves as a weighted mean of all inlier detections as the trajectory progresses. Similar to [36], we compare color models by their Bhattacharyya coefficient

$$p(a_i|\mathcal{A}) \sim \sum_q \sqrt{a_i(q)\mathcal{A}(q)}\ . \tag{9}$$

B. Dynamic Model.

Given a partially grown trajectory $\mathcal{H}_{t_0:t}$, we first select the subset of observations which fall inside its event cone. Using the following simple motion models

$$\begin{array}{lcl} \dot{x} = v\cos\theta & & \dot{x} = v\cos\theta \\ \dot{y} = v\sin\theta & \text{and} & \dot{y} = v\sin\theta \\ \dot{\theta} = K_c & & \dot{\theta} = K_c v \end{array} \tag{10}$$

for holonomic pedestrian and nonholonomic car motion on the ground plane, respectively, we compute predicted positions

$$\begin{array}{lcl} x^p_{t+1} = x_t + v\Delta t\cos\theta & & x^p_{t+1} = x_t + v\Delta t\cos\theta \\ y^p_{t+1} = y_t + v\Delta t\sin\theta & \text{and} & y^p_{t+1} = y_t + v\Delta t\sin\theta \\ \theta^p_{t+1} = \theta_t + K_c\Delta t & & \theta^p_{t+1} = \theta_t + K_c v\Delta t \end{array} \tag{11}$$

and approximate the positional uncertainty by an oriented Gaussian to arrive at the dynamic model $\mathcal{D}$

$$\mathcal{D}: \begin{array}{ll} p\left(\begin{bmatrix} x_{t+1} \\ y_{t+1} \end{bmatrix}\right) & \sim \mathcal{N}\left(\begin{bmatrix} x^p_{t+1} \\ y^p_{t+1} \end{bmatrix}, \Gamma^{\mathsf{T}}\begin{bmatrix} \sigma^2_{mov} & 0 \\ 0 & \sigma^2_{turn} \end{bmatrix}\Gamma\right) \\ p(\theta_{t+1}) & \sim \mathcal{N}(\theta^p_{t+1}, \sigma^2_{steer}) \end{array} . \tag{12}$$

Here, Γ is the rotation matrix, K_c the path curvature, and the nonholonomic constraint is approximated by adapting the rotational uncertainty σ_{turn} as a function of v.

C. Spacetime Trajectory Search for Moving Objects.

Each candidate observation $H_{i,t+1}$ is then evaluated under the covariance of $\mathcal{D}$ and compared to the trajectory's appearance model $\mathcal{A}$ (its mean color histogram), yielding

$$p(H_{i,t+1}|\mathcal{H}_{t_0:t}) = p(H_{i,t+1}|\mathcal{A}_t)p(H_{i,t+1}|\mathcal{D}_t). \tag{13}$$

After this, the trajectory is updated by the weighted mean of its predicted position and the supporting observations:

$$\mathbf{x}_{t+1} = \frac{1}{Z}\left(p(\mathcal{H}_{t:t+1}|\mathcal{H}_{t_0:t})\mathbf{x}^p_{t+1} + \sum_i p(H_{i,t+1}|\mathcal{H}_{t_0:t})\mathbf{x}_i\right), \tag{14}$$

with $p(\mathcal{H}_{t:t+1}|\mathcal{H}_{t_0:t}) = e^{-\lambda}$ and normalization factor Z. Velocity, rotation, and appearance model are updated in the same fashion. This process is iterated both forward and backward in time (Fig. 6(b)), and the resulting hypotheses are collected (Fig. 6(c)).

D. Temporal Accumulation for Static Objects.

Static objects are treated as a special case, since their sequence of prediction cones collapses to a spacetime cylinder with constant radius. For such a case, a more accurate localization estimate can be obtained by aggregating observations over a temporal window, which also helps to avoid localization jitter from inaccurate detections. Note that we do not have to make a decision whether an object is static or dynamic at this point. Instead, our system will typically create candidate hypotheses for both cases, leaving it to the model selection framework to select the one that better explains the data.

This is especially important for parked cars, since our appearance-based detectors provide a too coarse orientation to estimate a precise 3D bounding box. We therefore employ the method described in [8] for localization: the ground-plane locations of all detections within a time window are accumulated, and Mean-Shift mode estimation [5] is applied to accurately localize the hypothesis. For cars, we additionally estimate the orientation by fusing the orientation estimates from the single-view detectors with the principal axis of the cluster in a weighted average.

E. Global Trajectory Selection.

Taken together, the steps above result in a set of trajectory hypotheses for static and moving objects. It is important to point out that we do not prefer any of those hypotheses a priori. Instead, we let them compete in a hypothesis selection procedure in order to find the globally optimal explanation for the observed data. To this end, we express the support (or utility) $\mathcal{S}$ of a trajectory $\mathcal{H}_{t_0:t}$ reaching from time t_0 to t by the evidence collected from the images $I_{t_0:t}$ during that time span:

$$\begin{aligned} \mathcal{S}(\mathcal{H}_{t_0:t}|I_{t_0:t}) &= \sum_i \mathcal{S}(\mathcal{H}_{t_0:t}|H_{i,t_i})p(H_{i,t_i}|I_{t_i}) \\ &= p(\mathcal{H}_{t_0:t})\sum_i \frac{\mathcal{S}(H_{i,t_i}|\mathcal{H}_{t_0:t})}{p(H_{i,t_i})}p(H_{i,t_i}|I_{t_i}) \\ &\sim p(\mathcal{H}_{t_0:t})\sum_i \mathcal{S}(H_{i,t_i}|\mathcal{H}_{t_0:t})p(H_{i,t_i}|I_{t_i}), \end{aligned} \tag{15}$$

where $p(H_{i,t_i})$ is a normalization factor that can be omitted, since the later QBP stage enforces that each detection can only be assigned to a single trajectory. Further, we define

$$\begin{aligned} \mathcal{S}(H_{i,t_i}|\mathcal{H}_{t_0:t}) &= \mathcal{S}(\mathcal{H}_{t_i}|\mathcal{H}_{t_0:t})p(H_{i,t_i}|\mathcal{H}_{t_i}) \\ &= e^{-\lambda(t-t_i)}p(H_{i,t_i}|\mathcal{A}_{t_i})p(H_{i,t_i}|\mathcal{D}_{t_i})\ , \end{aligned} \tag{16}$$

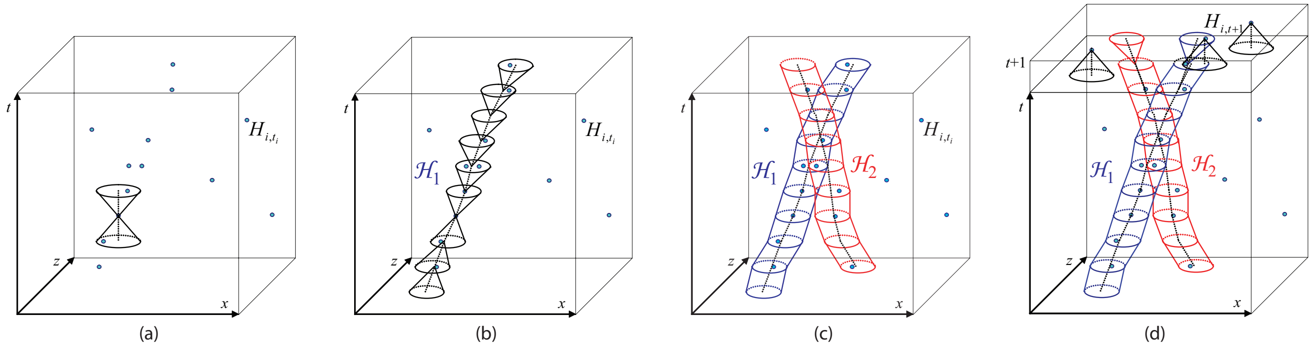
that is, we express the contribution of an observation H_{i,t_i} to trajectory $\mathcal{H}_{t_0:t} = (\mathcal{A}, \mathcal{D})_{t_0:t}$ by evaluating it under the trajectory's appearance and dynamic model at that time, weighted with a temporal discount.

In order to find the combination of trajectory hypotheses that together best explain the observed evidence, we again solve a Quadratic Boolean Problem $\max_m m^{\mathsf{T}}Qm$ with the additional constraint that no two objects may occupy the same space at the same time. With a similar derivation as in Section IV-A, we arrive at

$$\begin{aligned} q_{ii} &= -\epsilon_1 c(\mathcal{H}_{i,t_0:t}) + \sum_{H_{k,t_k}\in\mathcal{H}_i}\left((1-\epsilon_2) + \epsilon_2\ g_{k,i}\right) \\ q_{ij} &= -\frac{1}{2}\sum_{H_{k,t_k}\in\mathcal{H}_i\cap\mathcal{H}_j}\left((1-\epsilon_2) + \epsilon_2\ g_{k,\ell} + \epsilon_3\ O_{ij}\right) \\ g_{k,i} &= p^*(H_{k,t_k}|I_{t_k}) + \log p(H_{k,t_k}|\mathcal{H}_i), \end{aligned} \tag{17}$$

where $\mathcal{H}_\ell \in \{\mathcal{H}_i, \mathcal{H}_j\}$ denotes the weaker of the two trajectory hypotheses; $c(\mathcal{H}_{t_0:t}) \sim \#holes$ is a model cost that penalizes holes in the trajectory; and the additional penalty term O_{ij} measures the physical overlap between the spacetime trajectory volumes of $\mathcal{H}_i$ and $\mathcal{H}_j$ given average object dimensions.

Thus, two overlapping trajectory hypotheses compete both for supporting observations and for the physical space they occupy during their lifetime. This makes it possible to model complex object-object interactions, such that two pedestrians cannot walk through each other or that one needs to yield if the other shoves.



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

Fig. 6. *Visualization of the trajectory growing procedure. (a) Starting from an observation, we collect all detections that fall inside its event cone in the adjoining time steps and evaluate them under the trajectory model. (b) We adapt the trajectory based on inlier points and iterate the process both forward and backward 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 not built up from scratch at each time step, but are grown incrementally.*

The hypothesis selection procedure always searches for the best explanation of the current world state *given all evidence available up to now*. It is not guaranteed that this explanation is consistent with the one we got for the previous frame. However, as soon as it is selected, it explains the whole past, as if it had always existed. We can thus follow a trajectory back in time to determine where a pedestrian came from when he first stepped into view, even though no hypothesis was selected for him back then. Fig. 7 visualizes the estimated spacetime trajectories for such a case.

Although attractive in principle, this scheme needs to be made more efficient for practical applications, as explained next.

F. *Efficiency Considerations.*

The main computational cost in this stage comes from three factors: the cost to find trajectories, to build the quadratic interaction matrix Q, and to solve the final optimization problem. However, the first two steps can reuse information from previous time steps.

Thus, instead of building up trajectories from scratch at each time step t, we merely check for each of the existing hypotheses $\mathcal{H}_{t_0:t-k}$ if it can be extended by the new observations using eqs. (13) and (14). In addition, we start new trajectory searches down the time line from each new observation $H_{i,t-k+1:t}$, as visualized in Fig. 6(d). Note that this procedure does not require a detection in every frame; its time horizon can be set to tolerate large temporal gaps. Dynamic model propagation is unidirectional. After finding new evidence, the already existing part of the trajectory is not re-adjusted. However, in order to reduce the effect of localization errors, inevitably introduced by limitations of the object detector, the final trajectory hypothesis is smoothed by local averaging, and its score (15) is recomputed. Also note that most entries of the previous interaction matrix Q_{t-1} can be reused and just need to be weighted with the temporal discount $e^{-\lambda}$.

The optimization problem in general is NP-hard. In practice, the time required to find a good local maximum depends on the connectedness of the matrix Q, *i.e.* on the number of non-zero interactions between hypotheses. This number is typically very low for static objects, since only few hypotheses overlap. For pedestrian trajectories, the number of interactions may however grow quite large.

We use the multibranch gradient ascent method of [41], a simple local optimizer specifically designed for problems with high connectivity, but moderate number of variables and sparse solutions. In our experiments, it consistently outperforms not only simple greedy and Taboo search, but also the LP-relaxation of [4] (in computer vision also known as QPBO [40]), while branch-and-bound with the LP-relaxation as convex under-estimator has unacceptable computation times. Alternatives, which we have not tested but which we expect to perform similar to QPBO, are relaxations based on SDP and SOCP [23], [25].

VII. Coupled Detection & Trajectory Estimation.

As shown above, both object detection and trajectory estimation can be formulated as individual QBPs. However, the two tasks are closely coupled: the merit of a putative trajectory depends on the number and strength of the underlying detections $\{n_i = 1\}$, while the merit of a putative detection depends on the current object trajectories $\{m_i = 1\}$, which impose a prior on object locations. These dependencies lead to further interactions between detections and trajectories. In this section, we therefore jointly optimize both detections and trajectories by coupling them in a *combined* QBP.

However, we have to keep in mind that the relationship between detections and trajectories is not symmetric: trajectories ultimately rely on detections to be propagated, but new detections can occur without a trajectory to assign them to (*e.g.* when a new object enters the scene). In addition to the index vectors m for trajectories and n for detections, we therefore need to introduce a list of virtual trajectories v, one for each detection in the current image, to enable detections to survive without contributing to an actual trajectory. The effect of those virtual trajectories will be explained in detail in Sec. VII-A. We thus obtain the following joint optimization problem

$$\max_{m,v,n} \begin{bmatrix} m^{\mathsf{T}} & v^{\mathsf{T}} & n^{\mathsf{T}} \end{bmatrix} \begin{bmatrix} \widetilde{Q} & U & V \\ U^{\mathsf{T}} & R & W \\ V^{\mathsf{T}} & W^{\mathsf{T}} & \widetilde{S} \end{bmatrix} \begin{bmatrix} m \\ v \\ n \end{bmatrix}, \qquad (18)$$

where the elements of V, W model the interactions between detections and real and virtual trajectories, respectively, and U models the mutual exclusion between the two groups. The solution of (18) jointly optimizes both the detection results for the current frame, given the trajectories of the tracked objects, and the trajectories across frames, given the detections.

Equations (6) and (15) define the support that is used to build up our coupled optimization problem. This support is split up between the original matrices Q, S and the coupling matrices U, V, W as follows. The modified interaction matrix $\widetilde{Q}$ for the real trajectories keeps the form from (17), with the exception that

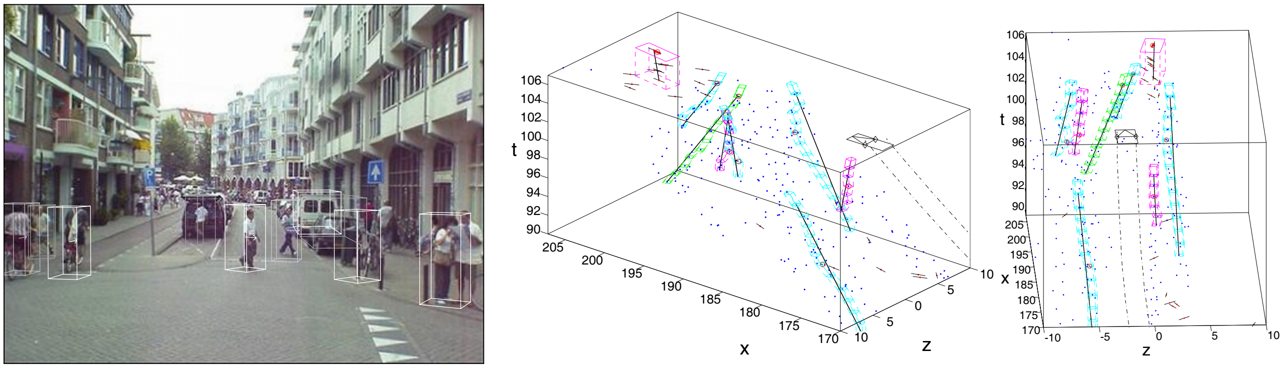


Fig. 7. *(left) Online 3D localization and trajectory estimation results of our system obtained from inside a moving vehicle. (The different bounding box intensities encode our system's confidence). (right) Visualizations of the corresponding spacetime trajectory estimates for this scene. Blue dots show pedestrian observations; red dots correspond to car observations.*

only the support from previous frames is entered into $\widetilde{Q}$:

$$\widetilde{q}_{ii} = -\epsilon_1 c(\mathcal{H}_{i,t_0:t}) + \sum_{H_{k,t_k}\in\mathcal{H}_{i,t_0:t-1}} \left((1-\epsilon_2)+\epsilon_2\ g_{k,i}\right) \tag{19}$$

$$\widetilde{q}_{ij} = -\frac{1}{2}\sum_{H_{k,t_k}\in(\mathcal{H}_i\cap\mathcal{H}_j)_{t_0:t-1}} \left((1-\epsilon_2)+\epsilon_2\ g_{k,\ell}+\epsilon_3\ O_{ij}\right). \tag{20}$$

The matrix R for the virtual trajectories contains simply the entries $r_{ii}=\varepsilon, r_{ij}=0$, with ε a very small constant. The matrix U for the interaction between real and virtual trajectories has entries u_{ik} that are computed similar to the real trajectory interactions q_{ij}

$$u_{ik} = -\frac{1}{2}\left((1-\epsilon_2)+\epsilon_2\ g_{k,i}+\epsilon_3\ O_{ik}\right). \tag{21}$$

The modified object detection matrix $\widetilde{S}$ contains as diagonal entries only the base cost of a detection, and as off-diagonal elements the full interaction cost between detections,

$$\widetilde{s}_{ii} = -\kappa_1\epsilon_2 - (1-\epsilon_2), \qquad \widetilde{s}_{ij} = s_{ij}. \tag{22}$$

Finally, the interaction matrices V, W between trajectories and detections have as entries the evidence a new detection contributes towards explaining the image data (which is the same as its contribution to a trajectory),

$$v_{ij} = \frac{1}{2}\left((1-\epsilon_2)+\epsilon_2 p^*(H_j|I_t)+\epsilon_2\log p(H_j|\mathcal{H}_i)\right) \tag{23}$$

$$w_{jj} = \max_i[v_{ij}]. \tag{24}$$

Note that R, S, and W are all quadratic and of the same size $N\times N$ and that R and W are diagonal matrices. As can be easily verified, the elements of the submatrices indeed add up to the correct objective function. Figure 8 visualizes the structure of the coupled optimization matrix.

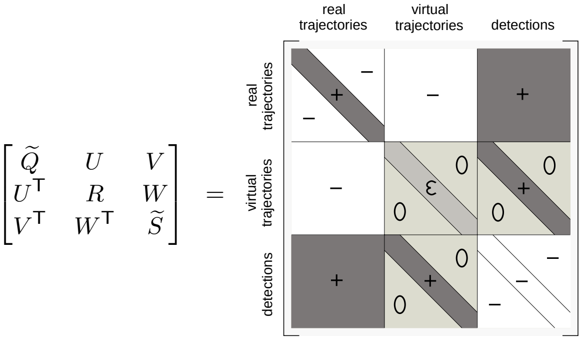


Fig. 8. *Structure of the coupled optimization matrix (eq. (18)).*

A. Discussion.

To illustrate this definition, we describe the most important features of the coupled optimization problem in words: 1) A trajectory is selected if its score outweighs the base cost in $\widetilde{q}_{ii}$. 2) If trajectory $\mathcal{H}_i$ is selected, and a compatible detection H_j is also selected, then H_j contributes to the trajectory score through v_{ij}. 3) If a detection H_j is not part of any trajectory, but its score outweighs the base cost in $\widetilde{s}_{jj}$, then it is still selected, with the help of its virtual trajectory and the contribution w_{jj}. 4) If a detection is part of any selected trajectory, then its virtual trajectory will not be selected, due to the interaction costs u_{ij} and the fact that the merit r_{jj} of a virtual trajectory is less than that of any real trajectory. 5) Finally, while all this happens, the detections compete for pixels in the image plane through the interaction costs $\widetilde{s}_{ij}$, and the trajectories compete for space in the object coordinate system through $\widetilde{q}_{ij}$.

Recapitulating the above, coupling has the following effects. First, it supports novel object detections that are consistent with existing trajectories. Eq. (23) states that existing trajectories impose a prior $p(H_j|\mathcal{H}_i)$ on certain object locations which raises the chance of generating novel detections there above the uniform background level $\mathcal{U}$. We model this prior as a Gaussian around the projected object position using the trajectory's dynamic model $\mathcal{D}$, so that $p(H_j|\{\mathcal{H}_i\})=\max[\mathcal{U},\max_i[\mathcal{N}(\mathbf{x}_i^p,\sigma_{pred}^2)]]$. Fig. 9 shows the prior for a frame from one of our test sequences. Second, the evidence from novel detections aids trajectories with which those detections are consistent by allowing them to account the new information as support.

B. Iterative Optimization.

Optimizing eq. (18) directly is difficult, since quadratic boolean optimization in its general form is NP hard. However, many QBPs obey additional simplifying constraints. In particular, the hypothesis selection problems for Q and S described earlier are submodular[3], and the expected solution is sparse (only few hypotheses will be selected), which allows one to find strong local maxima, as shown in [41]. However, the new QBP (18) is no longer submodular, since the interaction matrices V and W have positive entries.

We therefore resort to an EM-style iterative solution, which lends itself to the incremental nature of tracking: at each time

[3] Intuitively, submodularity is something like a discrete equivalent of convexity and means that the benefit of adding a certain element to a set can only decrease, but never increase, as the set grows.

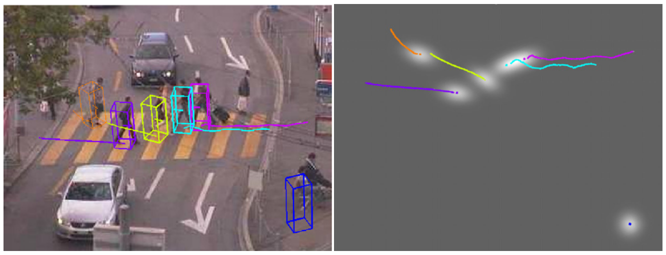


Fig. 9. *Influence of past trajectories on object detection.* Left: *25^{th} frame of sequence 2, and detected pedestrians.* Right: *Illustration of the detection prior for the 26^{th} frame. Top view showing trajectories estimated in the last frame, predicted positions, and detection prior (brighter color means higher probability).*

step t, object detection is solved using the trajectories from the previous frame $(t-1)$ as prior. In the above formulation, this corresponds to fixing the vector m. As an immediate consequence, we can split the detection hypotheses into two groups: those which are supported by a trajectory, and those which are not. We will denote the former by another binary index vector n^+, and the latter by its complement n^-. Since for fixed m the term $m^\top Qm = const.$, selecting detections amounts to solving

$$\begin{aligned}\max_{v,n}\left[\begin{bmatrix}v^\top & n^\top\end{bmatrix}\begin{bmatrix}R & W\\ W^\top & S\end{bmatrix}\begin{bmatrix}v\\ n\end{bmatrix} + 2m^\top\begin{bmatrix}U & V\end{bmatrix}\begin{bmatrix}v\\ n\end{bmatrix}\right] = \\ \max_{v,n}\begin{bmatrix}v^\top & n^\top\end{bmatrix}\begin{bmatrix}R+2\ \mathrm{diag}(U^\top m) & W\\ W^\top & S+2\ \mathrm{diag}(V^\top m)\end{bmatrix}\begin{bmatrix}v\\ n\end{bmatrix}.\end{aligned} \tag{25}$$

The interactions $U^\top m$ by construction only serve to suppress the virtual trajectories for the n^+. In contrast, $V^\top m$ adds the detection support from the n^+ to their score, while the diagonal interaction matrix W does the same for the n^-, which do not get their support through matrix V. We can hence further simplify to

$$\max_{n}\left[n^\top\left(R+S+2\ \mathrm{diag}(V^\top m)+2\ \mathrm{diag}(W^\top n^-)\right)n\right]. \tag{26}$$

The support W is only applied if no support comes from the trajectories and if in turn the interaction cost $U^\top m$ can be dropped, which only served to make sure W is outweighed for any n^+. The solution $\widehat{n}$ of (26) is the complete set of detections for the new frame; the corresponding virtual trajectories are $v = \widehat{n} \cap n^-$.

With the detection results from this step, the set of optimal trajectories is updated. This time, the detection results $[v^\top n^\top]$ are fixed, and the optimization reduces to

$$\max_{m}\left[m^\top\left(Q+2\ \mathrm{diag}(Vn)+2\ \mathrm{diag}(Uv)\right)m\right]. \tag{27}$$

The third term can be dropped, since virtual trajectories are now superseded by newly formed real trajectories. The second term is the contribution which the new detections make to the trajectory scores. The two reduced problems (26) and (27) are again submodular and can be solved with the multibranch ascent method of [41].

VIII. Implementation Details

The previous sections described the core components of our combined detection and tracking approach. However, as is often the case, several additional steps are required to guarantee good performance in practical applications.

A. Hypothesis Pruning.

Continually extending the existing hypotheses (while 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: candidates extrapolated through time for too long without finding any new evidence are removed. Similarly, 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 during optimization, 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.

B. Identity Management.

The hypothesis selection framework helps to ensure that all available information is used at each time step. However, it delivers an *independent explanation* at each time step and hence does not by itself keep track of object identities. Frame-to-frame propagation of tracked object identities is a crucial capability of tracking (as opposed to frame-by-frame detection).

Propagating identity is trivial in the case where a trajectory has been generated by extending one from the previous frame. In that case, the hypothesis ID is simply passed on, as in a recursive tracker. However, one of the core strengths of the presented approach is that it does not rely on stepwise trajectory extension alone. If at any time a newly generated hypothesis provides a better explanation for the observed evidence than an extended one, it will replace the older version. However, in this situation the new trajectory should inherit the old identity, in order to avoid an identity switch.

The problem can be solved with a simple strategy based on the associated data points: the identities of all selected trajectories are written into a buffer, together with the corresponding set of explained detections. This set is continuously updated as the trajectories grow. Each time a new trajectory is selected for the first time, it is compared to the buffer, and if its set of explained detections $\mathcal{E}_{\mathcal{H}} = \{H_i | H_i \in \mathcal{H}\}$ is similar to an entry $\mathcal{E}_{\mathcal{H}_k}$ in the buffer, it is identified as the new representative of that ID, replacing the older entry. If it does not match any known trajectory, it is added to the buffer with a new ID. For comparing the trajectory support, we use the following criterion:

$$\frac{|\mathcal{E}_{\mathcal{H}} \cap \mathcal{E}_{\mathcal{H}_k}|}{\min(|\mathcal{E}_{\mathcal{H}}|, |\mathcal{E}_{\mathcal{H}_k}|)} > \theta \quad \text{and} \quad k = \arg\max_{j} |\mathcal{E}_{\mathcal{H}} \cap \mathcal{E}_{\mathcal{H}_j}|. \tag{28}$$

C. Trajectory Initialization and Termination.

Object detection, together with the virtual trajectories introduced above, yields fully automatic track initialization. Given a new sequence, the system accumulates pedestrian detections in each new frame and tries to link them to detections from previous frames to obtain plausible spacetime trajectories, which are then fed into the selection procedure. After a few frames, the merit of a correct trajectory exceeds its cost, and an object track is started. Although several frames are required as evidence for a new track, the trajectory is in hindsight recovered from its beginning.

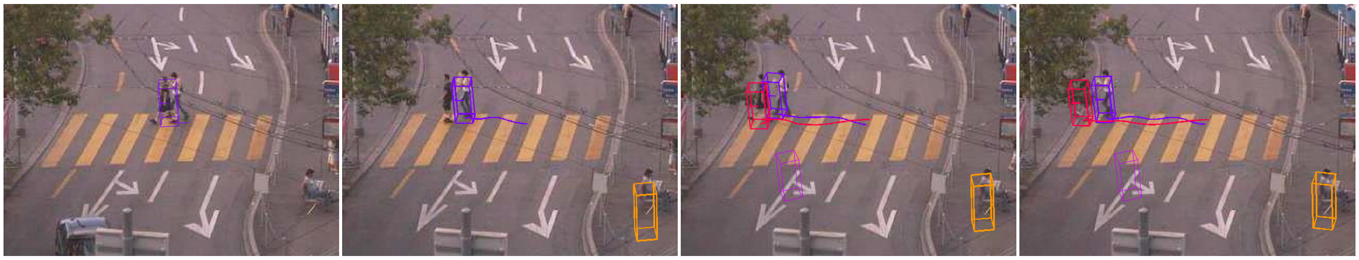


Fig. 10. *Example tracking results visualizing the non-Markovian nature of our approach. At the beginning of the sequence, both pedestrians walk close together and only one trajectory is initialized. However, when they separate sufficiently, a second trajectory is added that reaches back to the moment when both were first observed, while the first trajectory is automatically adjusted to make room for it.*

Algorithm 1 High-level overview of the tracking algorithm.

```
$\mathcal{H}_{\text{prev}} \leftarrow \emptyset$      // (all $\mathcal{H}$ without index $i$ denote sets of trajectories)
repeat
    Read current frame $I$, compute geometry $\mathcal{G}$.                    (Sec. III)

    // Create detection hypotheses                                  (Sec. V)
    Compute location priors $p(H)$, $p(H|\mathcal{H}_{\text{prev}})$.            (Sec. V-A,VII-A)
    $\{H_i\} \leftarrow$ getDetections($I, \mathcal{G}, p(H), p(H|\mathcal{H}_{\text{prev}})$)     (Sec. V-B)
    Build matrices $R, S, V, W$ using $\{H_i\}$ and $\mathcal{H}_{\text{prev}}$.      (Sec. VII)
    $\{H_{i,t}\} \leftarrow$ solve QBP($R, S, V, W$) from eq. (26).

    // Create trajectory hypotheses                                 (Sec. VI)
    $\mathcal{H}_{\text{extd}} \leftarrow$ extendTrajectories($\mathcal{H}_{\text{prev}}, \{H_{i,t}\}$)        (Sec. VI-F)
    $\mathcal{H}_{\text{stat}} \leftarrow$ growStaticTrajectories($\{H_{i,t_0:t}\}$)        (Sec. VI-C)
    $\mathcal{H}_{\text{dyn}} \leftarrow$ growDynamicTrajectories($\{H_{i,t_0:t}\}$)       (Sec. VI-D)
    $\mathcal{H}_{\text{all}} \leftarrow \{\mathcal{H}_{\text{end}}$, prune($\mathcal{H}_{\text{extd}}, \mathcal{H}_{\text{stat}}, \mathcal{H}_{\text{dyn}}$)$\}$    (Sec. VIII-A)
    Build matrices $Q, U, V$ using $\{H_{i,t}\}$ and $\mathcal{H}_{\text{all}}$.         (Sec. VII)
    $\mathcal{H}_{\text{acc}} \leftarrow$ solve QBP($Q, U, V$) from eq. (27).

    // Identity Management                                          (Sec. VIII-B)
    for all trajectories $\mathcal{H}_i \in \mathcal{H}_{\text{acc}}$ do
        Compare $\mathcal{H}_i$ with stored trajectories $\{\mathcal{H}_j\}$ and assign identity.

    // Check Termination                                            (Sec. VIII-C)
    for all trajectories $\mathcal{H}_i \in \mathcal{H}_{\text{acc}}$ do
        Check if $\mathcal{H}_i$ entered exit zone; if yes, move $\mathcal{H}_i$ to $\mathcal{H}_{\text{end}}$.

    // Propagate trajectories to next frame
    $\mathcal{H}_{\text{prev}} \leftarrow \mathcal{H}_{\text{all}} \setminus \mathcal{H}_{\text{end}}$.
until end of sequence.
```

The automatic initialization however means that trajectory termination needs to be handled explicitly: if an object leaves the scene, the detections along its track still exist and may prompt unwanted re-initializations. To control this behavior, exit zones are defined in 3D space along the image borders and are constantly monitored. When an object's trajectory enters the exit zone from within the image, the object is labeled as terminated, and its final trajectory is stored in a list of terminated tracks. To keep the tracker from re-using the underlying data, all trajectories from the termination list are added to the trajectory set and are always selected (inside a certain temporal window), thus preventing re-initializations based on the same detections through their interaction costs. The list of terminated tracks effectively serves as a memory, which ensures that the constraint that no two objects can occupy the same physical space at the same time survives after a hypothesis' termination. An overview of the complete tracking algorithm is shown in Alg. 1.

IX. Experimental Results

In the following, we evaluate our integrated approach on two challenging application scenarios. The first is a classical surveillance setting with a single camera monitoring a pedestrian crossing. Here, the task is to detect and track multiple pedestrians over long time frames and through occlusions.

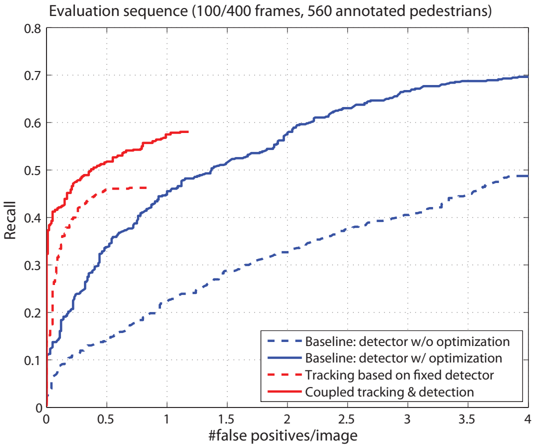


Fig. 11. *Performance comparison of our coupled detection+tracking system compared to various baselines.*

The second application scenario addresses the task of detecting and tracking other traffic participants from a moving vehicle. This task is considerably more difficult because of the combined effects of egomotion and a dynamically changing scene. On the other hand, each object will typically persist in the vehicle's field of view only for a few seconds. It is thus not as important to uniquely track a person's identity as in classic surveillance scenarios.

A. Tracking from a Static Surveillance Camera.

For tracking from a static surveillance camera, we demonstrate our approach on 3 test sequences. All sequences were recorded with a public webcam at 15fps, 320×240 pixels resolution, and contain severe MPEG compression artifacts. Note that a camera calibration is available for this setup, as the camera is static. In all result figures, line width denotes confidence of the recovered tracks: trajectories rendered with thin lines have lower scores.

Fig. 10 visualizes our approach's behavior on a short test sequence of two pedestrians crossing a street. At the beginning, they walk close together and the object detector often yields only a single detection. Thus, the support only suffices for a single trajectory to be initialized. However, as soon as the pedestrians separate, a second trajectory is instantiated that reaches back to the point at which both pedestrians were first observed. Together, the two trajectories provide a better explanation for the accumulated evidence and are therefore preferred by the model selection framework. As part of our optimization, both tracks are automatically adjusted such that their spacetime volumes do not intersect.

A more challenging case is displayed in Fig. 12. Here, multiple people cross the street at the same time, meeting in the middle. It

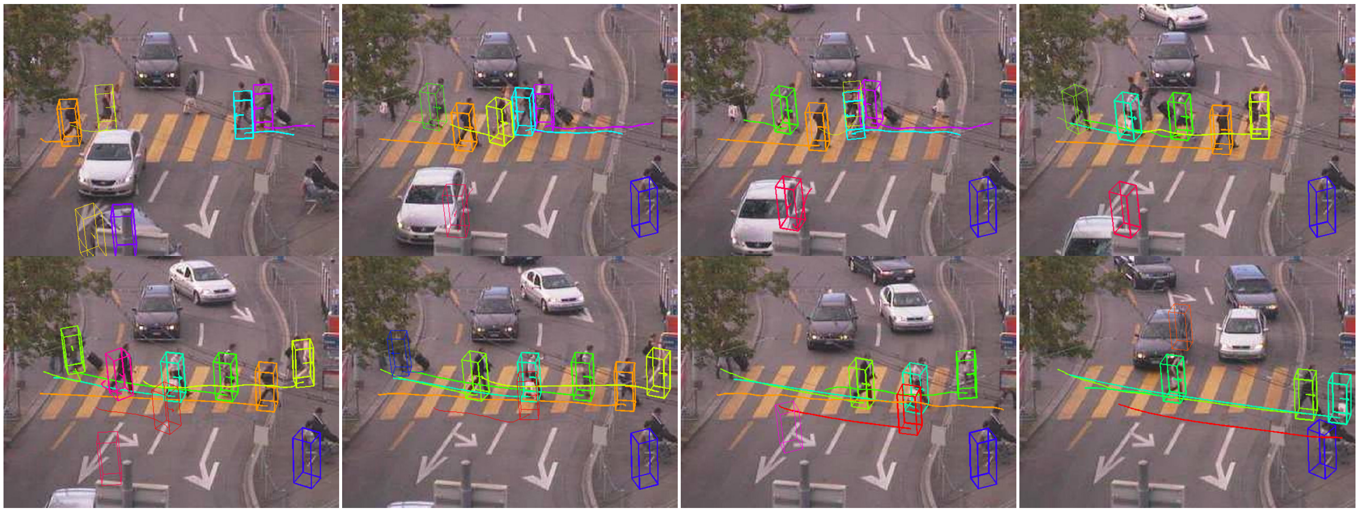


Fig. 12. *Tracking results on a pedestrian crossing scenario with occlusions and background changes.*

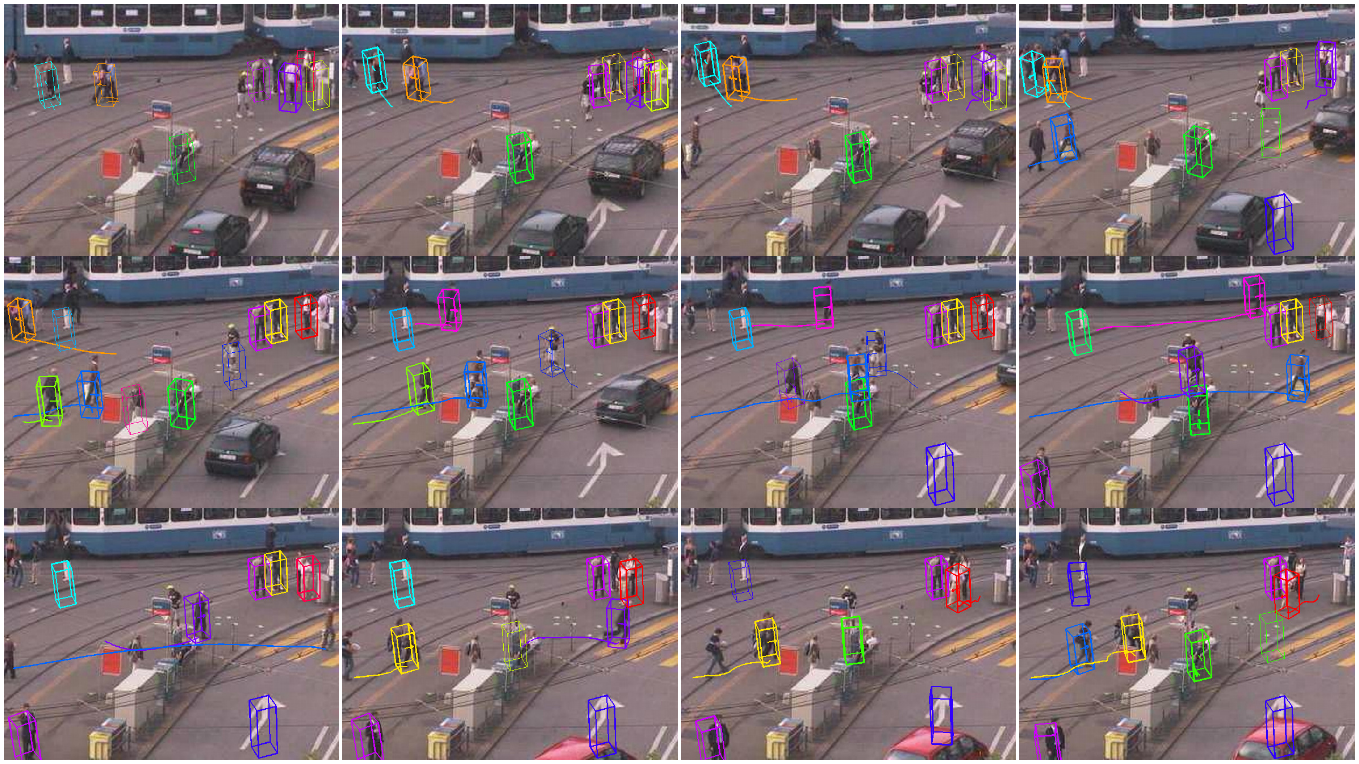


Fig. 13. *Results on a challenging sequence with many static pedestrians, frequent occlusions, and large-scale background changes.*

can be seen that, caused by the occlusion, our system temporarily loses track of two pedestrians, resulting in identity switches. However, it automatically recovers after few frames and returns to the correct identities. Again, this is something that classical Markovian tracking approaches are unable to do. In addition, our approach is able to detect and track the sitting person in the lower right corner which is indistinguishable from a static background. Relying on an object detector for input, we are however limited by the quality of the detections the latter can provide. Thus, our system will hypothesize wrong tracks in locations where the detector consistently produces false alarms.

For a quantitative assessment, we annotated every 4th frame of this sequence manually. We marked all image locations with 2D bounding boxes in which a person was visible. We then derived similar bounding boxes from the tracked 3D volumes and compared them to the annotations. Following recent object detection evaluations, we consider a box as correct if it overlaps with the ground-truth annotation by more than 50% using the intersection-over-union criterion [12]. Only one bounding box per annotation is counted as correct; every additional one is counted as a false positive. Note that this compares only localization accuracy, not person identities. Fig. 11 shows the result of our coupled system, compared to the baselines delivered by the object detector before and after QBP optimization (just matrix S) and to the baseline from a tracker based on fixed detections (decoupled matrices Q and S). Our approach improves on all three baselines and results in increased localization precision.

Finally, Fig. 13 presents results on a very challenging sequence with large-scale background changes from an incoming tram, many static pedestrians, and frequent occlusions. As can be seen from the result images, our system can track many of the pedestrians correctly over long periods despite these difficulties. Note especially the group of persons in the upper right image corner, which is correctly resolved throughout most of the sequence, as well as the pedestrian crossing the entire image width (shown in blue). The results confirm that our approach can deal with those difficulties and track its targets over long periods.

B. Tracking from a Moving Vehicle.

For this task, we evaluate our approach on two challenging video sequences. The first test sequence consists of 1175 image pairs recorded at 25fps and a resolution of 360×288 pixels over a distance of about 500m. It contains a total of 77 (sufficiently visible) static cars parked on both sides of the street, 4 moving cars, but almost no pedestrians at sufficiently high resolutions. The main difficulties for object detection here lie in the relatively low resolution, strong partial occlusion between parked cars, frequently encountered motion blur, and extreme contrast changes between brightly lit areas and dark shadows. Only the car detectors are used for this sequence.

The second sequence consists of 290 image pairs captured over the course of about 400m at the very sparse frame rate of 3fps and a resolution of 384×288 pixels. This very challenging sequence shows a vehicle passage through a crowded city center, with parked cars and bicycles on both street sides, numerous pedestrians and bicyclists travelling on the side walks and crossing the street, and several speed bumps. Apart from the difficulties mentioned above, this sequence poses the additional challenge of detecting and separating many mutually occluding pedestrians at very low resolutions while simultaneously limiting the number of false positives on background clutter. In addition, temporal integration is further complicated by the low frame rate.

In the following sections, we present experimental results for object detection and tracking performance on both sequences. However, it would clearly be unrealistic to expect perfect detection and tracking results under such difficult conditions, which may make the quantitative results hard to interpret. We therefore also provide the result videos at `http://www.vision.ethz.ch/bleibe/pami08`.

1) Object Detection Performance: Figure 14 displays example detection results of our system on difficult images from the two test sequences. All images have been processed at their original resolution by SfM and bilinearly interpolated to twice their initial size for object detection. For a quantitative evaluation we annotated one video stream for each sequence and marked all objects that were within 50m distance and visible by at least 30-50%. It is important to note that this includes many cases with partial visibility. Fig 15(left) shows the resulting detection performance with and without ground plane constraints. As can be seen from the plots, both recall and precision are greatly improved by the inclusion of scene geometry, up to an operating point of 0.34 fp/frame at 46-47% recall for cars and 1.65 fp/frame at 42% recall for pedestrians.

In order to put those results into perspective, Fig. 15(right) shows a detailed evaluation of the recognition performance as a function of the object distance (as obtained from the groundplane estimate). As can be seen from those plots, both the car and pedestrian detectors perform best up to a distance of 25-30m, after which recall drops off. Consequently, both precision and recall are notably improved when only considering objects up to a distance of 25m (as again shown in Fig. 15(left)).

For cars, the distribution of false positives over distances follows the distribution of available objects (shown in Fig. 15(middle)), indicating that most false positives are indeed caused by car structures (which is also consistent with our visual impression). For pedestrians, it can be observed that most false positives occur at closer scales. This can be explained by the presence of extremely cluttered regions (*e.g.* bike racks) in the second sequence and by the fact that many closer pedestrians are only partially visible behind parked cars.

2) Tracking Performance: Figure 16 shows online tracking results of our system (using only detections from previous frames) for both sequences. As can be seen, our system manages to localize and track other traffic participants despite significant egomotion and dynamic scene changes. The 3D localization and orientation estimates typically converge at a distance of 15-30m and lead to accurate 3D bounding boxes for cars and pedestrians. A major challenge for sequence #2 is to filter out false positives from incorrect detections. At 3fps, this is not always possible. However, false positives typically get only low confidence ratings and quickly fade out again as they fail to get continuous support.

X. Conclusion

In this paper, we have presented a novel approach for multi-object tracking that couples object detection and trajectory estimation in a combined model selection framework. Our approach does not rely on a Markov assumption, but can integrate information over long time periods to revise its decision and recover from mistakes in the light of new evidence. As our approach is based on continuous detection, it can operate with both static and moving cameras and cope with large-scale background changes.

We have applied this method to build an integrated system for dynamic 3D scene analysis from a moving platform. The resulting system fuses the output of multiple single-view object detectors and integrates continuously reestimated scene geometry constraints. Together with an online calibration from SfM, it aggregates detections over time to accurately localize and track a large and variable number of objects in difficult scenes. As our experiments demonstrate, the proposed approach is able to obtain an accurate analysis of dynamic scenes, even at low frame rates.

A current limitation is the overall run-time of the approach. Although many of the presented steps run at several frames per second, the system as a whole is not yet capable of real-time performance in our current implementation. We are currently working on speedups to remedy this issue. Also, since the tracking framework operates in 3D, it is constrained to scenarios where either a camera calibration or SfM can be robustly obtained.

In this paper, we have focused on tracking pedestrians and cars. This can be extended to other object categories for which reliable object detectors are available [12]. Also, we want to point out that our approach is not restricted to the ISM detector. It can be applied based on any detector that performs sufficiently well, such as *e.g.* the detectors by [45] or [10] (in the latter case taking an approximation for the top-down segmentation). Other possible extensions include the integration of additional cues such as stereo depth [11], or the combination with adaptive background modeling for static cameras.

Acknowledgments: This work is supported, in parts, by EU projects DIRAC (IST-027787) and HERMES (IST-027110). We also acknowledge the support

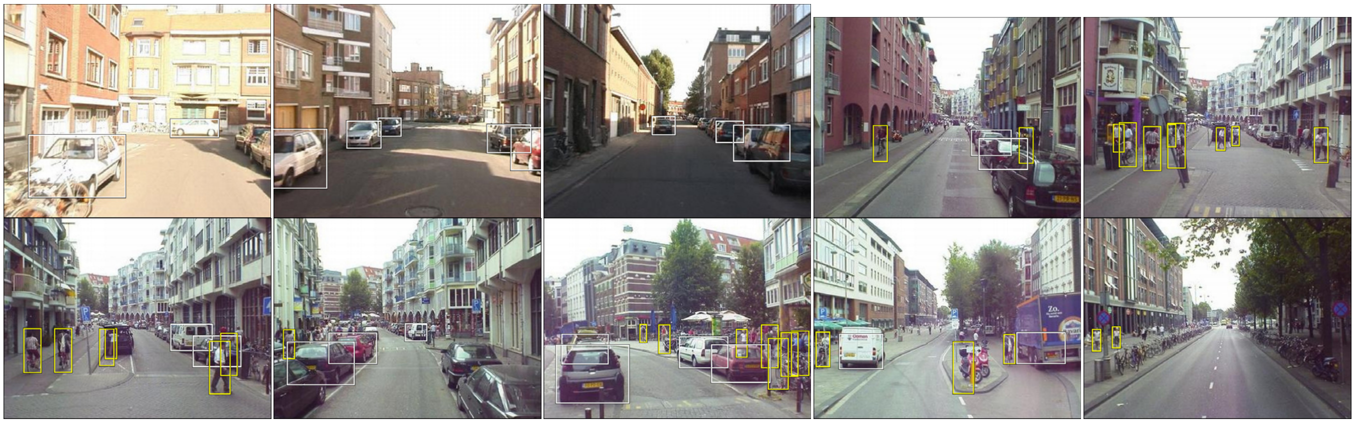


Fig. 14. *Example car and pedestrian detections of our system on difficult images from the two test sequences.*

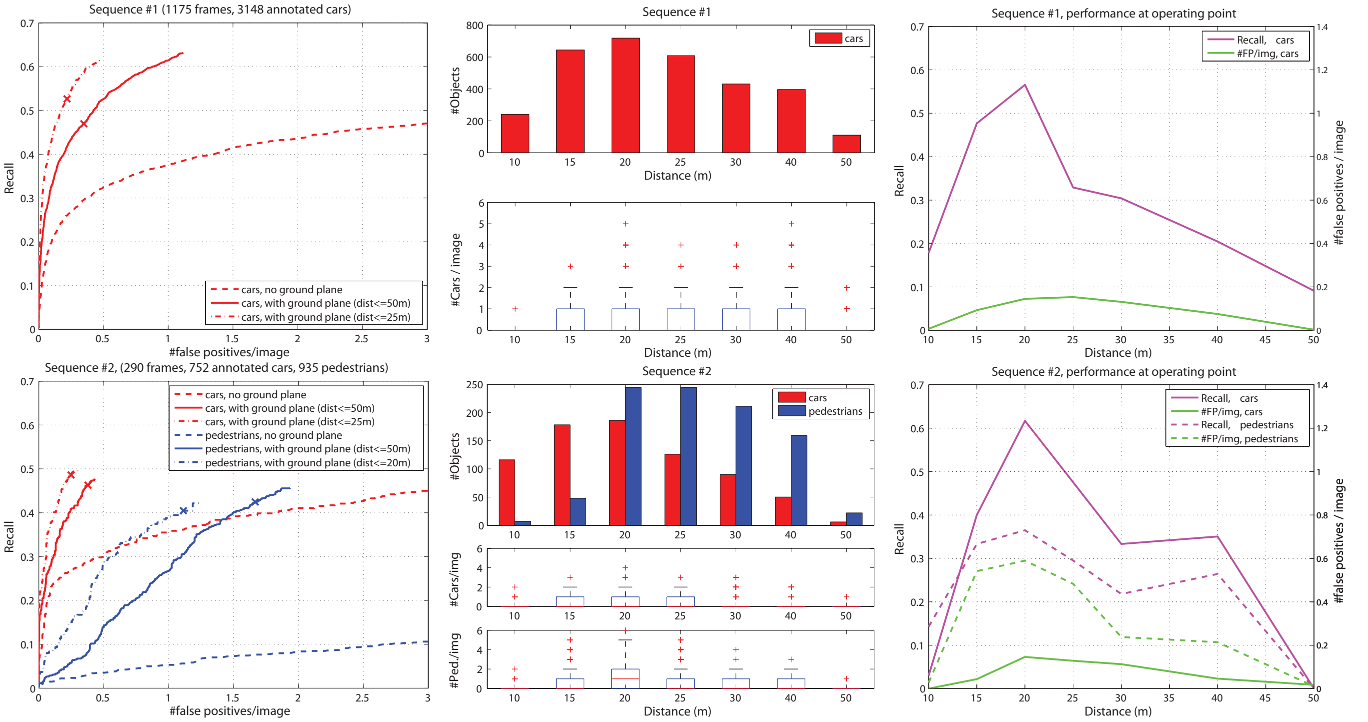


Fig. 15. *(left) Quantitative comparison of the detection performance with and without scene geometry constraints (the crosses mark the operating point for tracking). (middle) Absolute and average number of annotated objects as a function of their distance. (right) Detection performance as a function of the object distance.*

of Toyota Motor Corporation/Toyota Motor Europe and KU Leuven Research Fund's GOA project MARVEL. We thank TeleAtlas for providing additional survey videos to test on and K. Cornelis for his work on the SfM system.

References

[1] S. Avidan. Ensemble tracking. In *CVPR'05*, 2005.

[2] J. Berclaz, F. Fleuret, and P. Fua. Robust people tracking with global trajectory optimization. In *CVPR'06*, pages 744–750, 2006.

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

[4] E. Boros and P. Hammer. Pseudo-boolean optimization. *Discrete Applied Mathematics*, 123(1-3):155–225, 2002.

[5] D. Comaniciu and P. Meer. Mean Shift: A robust approach toward feature space analysis. *PAMI*, 24(5):603–619, 2002.

[6] D. Comaniciu, V. Ramesh, and P. Meer. Kernel-based object tracking. *PAMI*, 25(5):564–575, 2003.

[7] N. Cornelis, K. Cornelis, and L. Van Gool. Fast compact city modeling for navigation pre-visualization. In *CVPR'06*, 2006.

[8] N. Cornelis, B. Leibe, K. Cornelis, and L. Van Gool. 3D urban scene modeling integrating recognition and reconstruction. *IJCV*, 78(2-3):121–141, 2008.

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

[10] N. Dalal and B. Triggs. Histograms of oriented gradients for human detection. In *CVPR'05*, 2005.

[11] A. Ess, B. Leibe, and L. Van Gool. Depth and appearance for mobile scene analysis. In *ICCV'07*, 2007.

[12] M. Everingham and others (34 authors). The 2005 PASCAL Visual Object Class Challenge. In *Machine Learning Challenges. Evaluating Predictive Uncertainty, Visual Object Classification, and Recognising Textual Entailment*, LNAI 3944. Springer, 2006.

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

[14] D. Gavrila and V. Philomin. Real-time object detection for smart vehicles. In *ICCV'99*, pages 87–93, 1999.

[15] A. Gelb. *Applied Optimal Estimation*. MIT Press, 1996.

[16] J. Giebel, D. Gavrila, and C. Schnörr. A Bayesian framework for multi-cue 3D object tracking. In *ECCV'04*, 2004.

[17] H. Grabner and H. Bischof. On-line boosting and vision. In *CVPR'06*, pages 260–267, 2006.

[18] R. Hartley and A. Zisserman. *Multiple view geometry in computer vision*. Cambridge University Press, 2000.

[19] D. Hoiem, A. Efros, and M. Hebert. Geometric context from a single

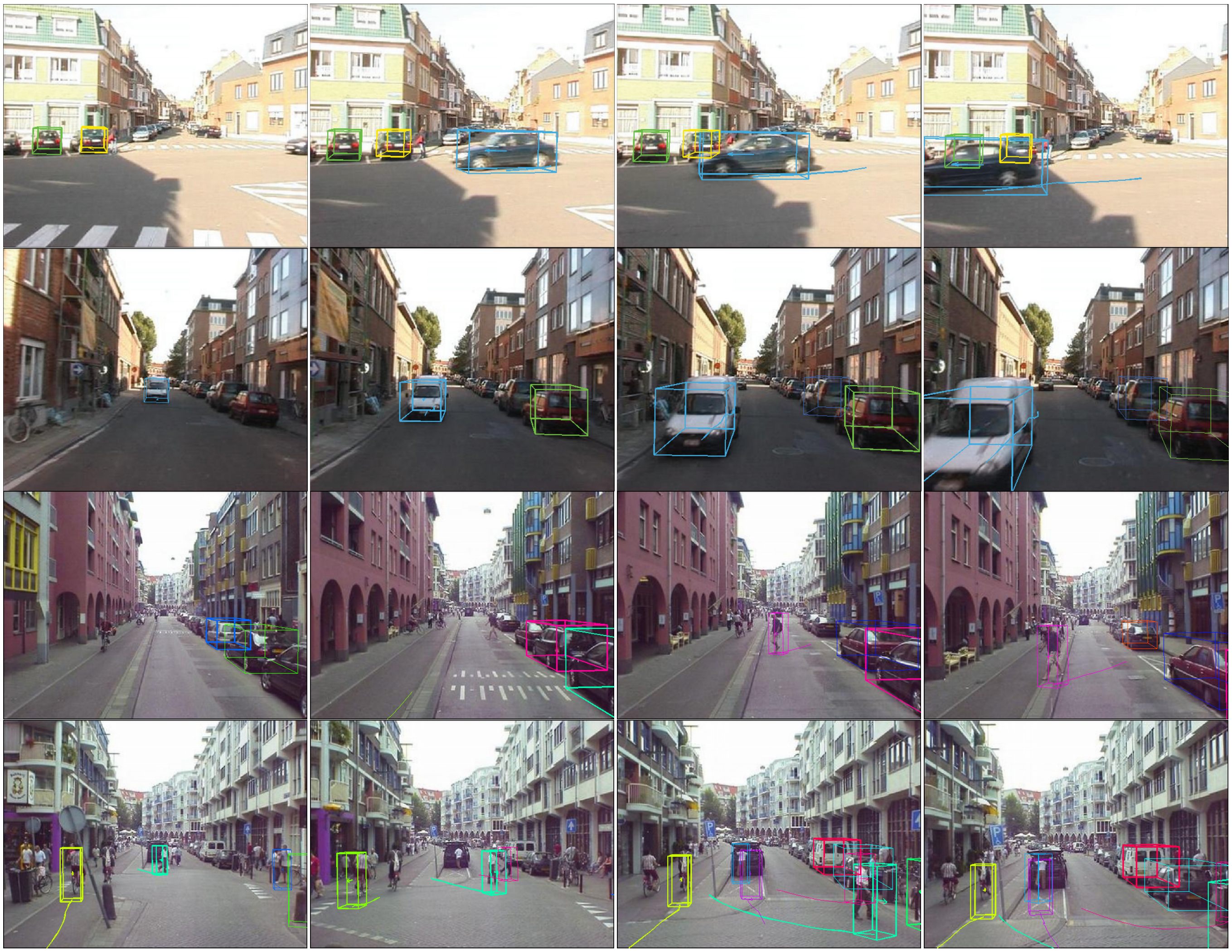


Fig. 16. *3D localization and tracking results of our system from inside a moving vehicle.*

image. In *ICCV'05*, 2005.

[20] D. Hoiem, A. Efros, and M. Hebert. Putting objects into perspective. In *CVPR'06*, 2006.

[21] M. Isard and A. Blake. CONDENSATION–conditional density propagation for visual tracking. *IJCV*, 29(1), 1998.

[22] R. Kaucic, A. Perera, G. Brooksby, J. Kaufhold, and A. Hoogs. A unified framework for tracking through occlusions and across sensor gaps. In *CVPR'05*, 2005.

[23] J. Keuchel. Multiclass image labeling with semidefinite programming. In *ECCV'06*, pages 454–467, 2006.

[24] D. Koller, K. Daniilidis, and H.-H. Nagel. Model-based object tracking in monocular image sequences of road traffic scenes. *IJCV*, 10(3):257–281, 1993.

[25] M. Kumar, P. Torr, and A. Zisserman. Solving Markov random fields using second order cone programming relaxations. In *CVPR'06*, 2006.

[26] O. Lanz. Approximate Bayesian multibody tracking. *PAMI*, 28(9):1436–1449, 2006.

[27] B. Leibe, N. Cornelis, K. Cornelis, and L. Van Gool. Dynamic 3D scene analysis from a moving vehicle. In *CVPR'07*, 2007.

[28] B. Leibe, A. Leonardis, and B. Schiele. Robust object detection with interleaved categorization and segmentation. *IJCV*, 77(1-3):259–289, 2008.

[29] B. Leibe, K. Mikolajczyk, and B. Schiele. Segmentation based multi-cue integration for object detection. In *BMVC'06*, 2006.

[30] B. Leibe, K. Schindler, and L. Van Gool. Coupled detection and trajectory estimation for multi-object tracking. In *ICCV'07*, 2007.

[31] B. Leibe, E. Seemann, and B. Schiele. Pedestrian detection in crowded scenes. In *CVPR'05*, 2005.

[32] A. Leonardis, A. Gupta, and R. Bajcsy. Segmentation of range images as the search for geometric parametric models. *IJCV*, 14:253–277, 1995.

[33] D. Lowe. Distinctive image features from scale-invariant keypoints. *IJCV*, 60(2):91–110, 2004.

[34] K. Mikolajczyk, B. Leibe, and B. Schiele. Multiple object class detection with a generative model. In *CVPR'06*, 2006.

[35] K. Mikolajczyk and C. Schmid. A performance evaluation of local descriptors. *PAMI*, 27(10), 2005.

[36] K. Nummiaro, E. Koller-Meier, and L. Van Gool. An adaptive color-based particle filter. *Image and Vision Computing*, 21(1):99–110, 2003.

[37] K. Okuma, A. Taleghani, N. de Freitas, J. Little, and D. Lowe. A boosted particle filter: Multitarget detection and tracking. In *ECCV'04*, 2004.

[38] V. Philomin, R. Duraiswami, and L. Davis. Pedestrian tracking from a moving vehicle. In *Intel. Vehicles Symp.'00*, pages 350–355, 2000.

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

[40] C. Rother, V. Kolmogorov, V. Lempitsky, and M. Szummer. Optimizing binary MRFs via extended roof duality. In *CVPR'07*, 2007.

[41] K. Schindler, J. U, and H. Wang. Perspective n-view multibody structure-and-motion through model selection. In *ECCV'06*, pages 606–619, 2006.

[42] C. Stauffer and W. Grimson. Adaptive background mixture models for realtime tracking. In *CVPR'99*, 1999.

[43] P. Viola and M. Jones. Robust real-time face detection. *IJCV*, 57(2):137–154, 2004.

[44] P. Viola, M. Jones, and D. Snow. Detecting pedestrians using patterns of motion and appearance. In *ICCV'03*, pages 734–741, 2003.

[45] B. Wu and R. Nevatia. Detection of multiple, partially occluded humans in a single image by Bayesian combination of edgelet part detectors. In *ICCV'05*, 2005.

[46] B. Wu and R. Nevatia. Tracking of multiple, partially occluded humans based on static body part detections. In *CVPR'06*, 2006.

[47] F. Yan, A. Kostin, W. Christmas, and J. Kittler. A novel data association algorithm for object tracking in clutter with application to tennis video analysis. In *CVPR'06*, 2006.