3D Vehicle Trajectory Reconstruction in Monocular Video Data Using Environment Structure Constraints

Sebastian Bullinger[1][0000−0002−1584−5319], Christoph Bodensteiner[1], Michael Arens[1], and Rainer Stiefelhagen[2]

[1] Fraunhofer IOSB, Ettlingen, Germany
{sebastian.bullinger,christoph.bodensteiner,michael.arens}
@iosb.fraunhofer.de
[2] Karlsruhe Institute of Technology, Karlsruhe, Germany
rainer.stiefelhagen@kit.edu

Abstract. We present a framework to reconstruct three-dimensional vehicle trajectories using monocular video data. We track two-dimensional vehicle shapes on pixel level exploiting instance-aware semantic segmentation techniques and optical flow cues. We apply Structure from Motion techniques to vehicle and background images to determine for each frame camera poses relative to vehicle instances and background structures. By combining vehicle and background camera pose information, we restrict the vehicle trajectory to a one-parameter family of possible solutions. We compute a ground representation by fusing background structures and corresponding semantic segmentations. We propose a novel method to determine vehicle trajectories consistent to image observations and reconstructed environment structures as well as a criterion to identify frames suitable for scale ratio estimation. We show qualitative results using drone imagery as well as driving sequences from the Cityscape dataset. Due to the lack of suitable benchmark datasets we present a new dataset to evaluate the quality of reconstructed three-dimensional vehicle trajectories. The video sequences show vehicles in urban areas and are rendered using the path-tracing render engine Cycles. In contrast to previous work, we perform a quantitative evaluation of the presented approach. Our algorithm achieves an average reconstruction-to-ground-truth-trajectory distance of 0.31 meter using this dataset. The dataset including evaluation scripts will be publicly available on our website[3].

Keywords: Vehicle Trajectory Reconstruction, Instance-aware Semantic Segmentation, Structure-from-Motion

1 Introduction

1.1 Trajectory Reconstruction

Three-dimensional vehicle trajectory reconstruction has many relevant use cases in the domain of autonomous systems and augmented reality applications. There

[3] Project page: http://s.fhg.de/trajectory

are different platforms like drones or wearable systems where one wants to achieve this task with a minimal number of devices in order to reduce weight or lower production costs. We propose a novel approach to reconstruct three-dimensional vehicle motion trajectories using a single camera as sensor.
The reconstruction of object motion trajectories in monocular video data captured by moving cameras is a challenging task, since in general it cannot be solely solved exploiting image observations. Each observed object motion trajectory is scale ambiguous. Additional constraints are required to identify a motion trajectory consistent to environment structures. [26,14,3] assume that the camera is mounted on a driving vehicle, i.e. the camera has specific height and a known pose. [18,31,17,19] solve the scale ambiguity by making assumptions about object and camera motion trajectories. We follow Ozden's principle of non-accidental motion trajectories [18] and introduce a new object motion constraint exploiting semantic segmentation and terrain geometry to compute consistent object motion trajectories.
In many scenarios, objects cover only a minority of pixels in video frames. This increases the difficulty of reconstructing object motion trajectories using image data. In such cases, current state-of-the-art Structure from Motion (SfM) approaches treat vehicle observations most likely as outliers and reconstruct background structures instead. Previous works, e.g. [12,13], tackle this problem by considering multiple video frames to determine moving parts in the video. They apply motion segmentation or keypoint tracking to detect moving objects. These kind of approaches are vulnerable to occlusion and require objects to move in order to separate them from background structures.
Our method exploits recent results in instance-aware semantic segmentation and rigid Structure from Motion techniques. Thus, our approach extends naturally to stationary vehicles. In addition, we do not exploit specific camera pose constraints like a fixed camera-ground-angle or a fixed camera-ground-distance. We evaluate the presented vehicle trajectory reconstruction algorithm in UAV scenarios, where such constraints are not valid.

1.2 Related Work

Semantic segmentation or scene parsing is the task of providing semantic information at pixel-level. Early semantic segmentation approaches using ConvNets, e.g. Farabet et al. [6], exploit patchwise training. Long et al. [24] applied Fully Convolutional Networks for semantic segmentation, which are trained end-to-end. Recently, [5,15,10] proposed instance-aware semantic segmentation approaches.
The field of Structure from Motion (SfM) can be divided into iterative and global approaches. Iterative or sequential SfM methods [25,30,16,27,23] are more likely to find reasonable solutions than global SfM approaches [16,27]. However, the latter are less prone to drift.
The determination of the correct scale ratio between object and background reconstruction requires additional constraints. Ozden et al. [18] exploit the non-accidentalness principle in the context of independently moving objects. Yuan et

al. [31] propose to reconstruct the 3D object trajectory by assuming that the object motion is perpendicular to the normal vector of the ground plane. Kundu et al. [12] exploit motion segmentation with multibody VSLAM to reconstruct the trajectory of moving cars. They use an instantaneous constant velocity model in combination with Bearing only Tracking to estimate consistent object scales. Park et al. propose an approach in [19] to reconstruct the trajectory of a single 3D point tracked over time by approximating the motion using a linear combination of trajectory basis vectors. Previous works, like [18,31,12,19] show only qualitative results.

1.3 Contribution

The core contributions of this work are as follows. (1) We present a new framework to reconstruct the three-dimensional trajectory of vehicles in monocular video data leveraging state-of-the-art semantic segmentation and structure from motion approaches. (2) We propose a novel method to compute vehicle motion trajectories consistent to image observations and environment structures including a criterion to identify frames suitable for scale ratio estimation. (3) In contrast to previous work, we quantitatively evaluate the reconstructed vehicle motion trajectories. (4) We created a new vehicle trajectory benchmark dataset due to the lack of publicly available video data of vehicles with suitable ground truth data. The dataset consists of photo-realistic rendered videos of urban environments. It includes animated vehicles as well as set of predefined camera and vehicle motion trajectories. 3D vehicle and environmental models used for rendering serve as ground truth. (5) We will publish the dataset and evaluation scripts to foster future object motion reconstruction related research.

1.4 Paper Overview

The paper is organized as follows. Section 2 describes the structure and the components of the proposed pipeline. In section 2.1 we derive an expression for a one-parameter family of possible vehicle motion trajectories combining vehicle and background reconstruction results. Section 2.2 describes a method to approximate the ground locally. In section 2.3 we describe a method to compute consistent vehicle motion trajectories. In section 4 we provide an qualitative and quantitative evaluation of the presented algorithms using driving sequences, drone imagery and rendered video data. Section 5 concludes the paper.

2 Object Motion Trajectory Reconstruction

Fig. 1 shows the elements of the proposed pipeline. We use the approach presented in [2] to track two-dimensional vehicle shapes in the input video on pixel level. We detect vehicle shapes exploiting the instance-aware semantic segmentation method presented in [15] and associate extracted object shapes of subsequent frames using the optical flow approach described in [11]. Without loss of

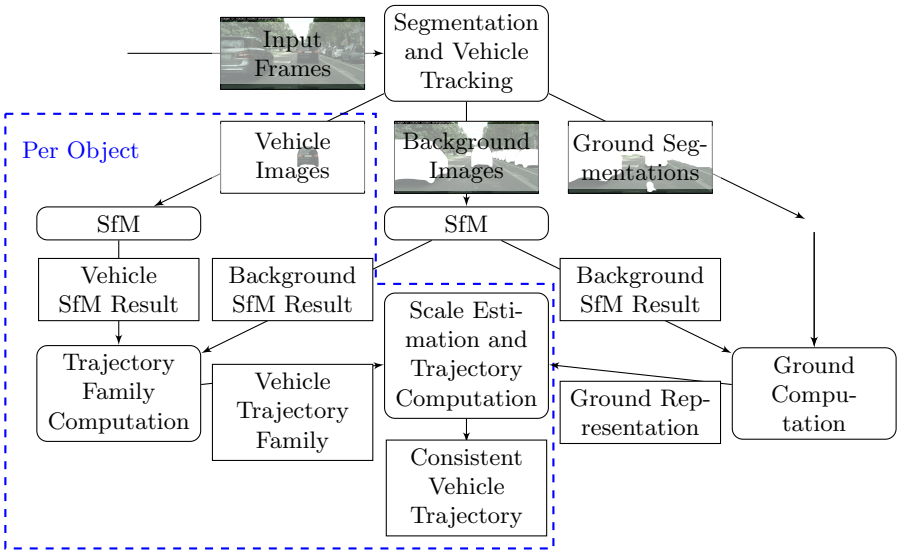


Fig. 1: Overview of the trajectory reconstruction pipeline. Boxes with corners denote computation results and boxes with rounded corners denote computation steps, respectively.

generality, we describe motion trajectory reconstructions of single objects. We apply SfM [16,23] to object and background images as shown in Fig. 1. Object images denote images containing only color information of single object instance. Similarly, background images show only background structures. We combine object and background reconstructions to determine possible, visually identical, object motion trajectories. We compute a consistent object motion trajectory exploiting constraints derived from reconstructed terrain ground geometry.

2.1 Object Trajectory Representation

In order to estimate a consistent object motion trajectory we apply SfM simultaneously to vehicle/object and background images as shown in Fig. 1. We denote the corresponding SfM results with $sfm^{(o)}$ and $sfm^{(b)}$. Let $\mathbf{o}_j^{(o)} \in \mathcal{P}^{(o)}$ and $\mathbf{b}_k^{(b)} \in \mathcal{P}^{(b)}$ denote the 3D points contained in $sfm^{(o)}$ or $sfm^{(b)}$, respectively. The superscripts o and b in $\mathbf{o}_j^{(o)}$ and $\mathbf{b}_k^{(b)}$ describe the corresponding coordinate frame. The variables j and k are the indices of points in the object or the background point cloud, respectively. We denote the reconstructed intrinsic and extrinsic parameters of each registered input image as virtual camera. Each virtual camera in $sfm^{(o)}$ and $sfm^{(b)}$ corresponds to a certain frame from which object and background images are extracted. We determine pairs of corresponding virtual cameras contained in $sfm^{(o)}$ and $sfm^{(b)}$. In the following, we consider only camera pairs, whose virtual cameras are contained in $sfm^{(o)}$

and $sfm^{(b)}$. Because of missing image registrations this may not be the case for all virtual cameras.
We reconstruct the object motion trajectory by combining information of corresponding virtual cameras. Our method is able to determine the scale ratio using a single camera pair. For any virtual camera pair of an image with index i the object SfM result $sfm^{(o)}$ contains information of object point positions $\mathbf{o}_j^{(o)}$ relative to virtual cameras with camera centers $\mathbf{c}_i^{(o)}$ and rotations $\mathbf{R}_i^{(o)}$. We express each object point $\mathbf{o}_j^{(o)}$ in camera coordinates $\mathbf{o}_j^{(i)}$ of camera i using $\mathbf{o}_j^{(i)} = \mathbf{R}_i^{(o)} \cdot (\mathbf{o}_j^{(o)} - \mathbf{c}_i^{(o)})$. The background SfM result $sfm^{(b)}$ contains the camera center $\mathbf{c}_i^{(b)}$ and the corresponding rotation $\mathbf{R}_i^{(b)}$, which provide pose information of the camera with respect to the reconstructed background. Note that the camera coordinate systems of virtual cameras in $sfm^{(o)}$ and $sfm^{(b)}$ are equivalent. We use $\mathbf{c}_i^{(b)}$ and $\mathbf{R}_i^{(b)}$ to transform object points to the background coordinate system using $\mathbf{o}_{j,i}^{(b)} = \mathbf{c}_i^{(b)} + {\mathbf{R}_i^{(b)}}^T \cdot \mathbf{o}_j^{(i)}$. In general, the scale ratio of object and background reconstruction does not match due to the scale ambiguity of SfM reconstructions [9]. We tackle this problem by treating the scale of the background as reference scale and by introducing a scale ratio factor r to adjust the scale of object point coordinates. The overall transformation of object points given in object coordinates $\mathbf{o}_j^{(o)}$ to object points in the background coordinate frame system $\mathbf{o}_{j,i}^{(b)}$ of camera i is described according to equation (1).

$$\mathbf{o}_{j,i}^{(b)} = \mathbf{c}_i^{(b)} + r \cdot {\mathbf{R}_i^{(b)}}^T \cdot \mathbf{R}_i^{(o)} \cdot (\mathbf{o}_j^{(o)} - \mathbf{c}_i^{(o)}) \coloneqq \mathbf{c}_i^{(b)} + r \cdot \mathbf{v}_{j,i}^{(b)} \tag{1}$$

with

$$\mathbf{v}_{j,i}^{(b)} = {\mathbf{R}_i^{(b)}}^T \cdot \mathbf{R}_i^{(o)} \cdot (\mathbf{o}_j^{(o)} - \mathbf{c}_i^{(o)}) = \mathbf{o}_{j,i}^{(b)} - \mathbf{c}_i^{(b)}. \tag{2}$$

Given the scale ratio r, we can recover the full object motion trajectory computing equation (2) for each virtual camera pair. We use $\mathbf{o}_{j,i}^{(b)}$ of all cameras and object points as object motion trajectory representation. The ambiguity mentioned in section 1 is expressed by the unknown scale ratio r.

2.2 Terrain Ground Approximation

Further camera or object motion constraints are required to determine the scale ratio r introduced in equation (2). In contrast to previous work [18,31,19,14,26,3] we assume that the object category of interest moves on top of the terrain. We exploit semantic segmentation techniques to estimate an approximation of the ground surface of the scene. We apply the ConvNet presented in [24] to determine ground categories like street or grass for all input images on pixel level. We consider only stable background points, i.e. 3D points that are observed at least four times. We determine for each 3D point a ground or non-ground label by accumulating the semantic labels of corresponding keypoint measurement pixel positions. This allows us to determine a subset of background points, which represent the ground of the scene. We approximate the ground surface locally using

plane representations. For each frame i we use corresponding estimated camera parameters and object point observations to determine a set of ground points P_i close to the object. We build a kd-tree containing all ground measurement positions of the current frame. For each object point observation, we determine the num_b closest background measurements. In our experiments, we set num_b to 50. Let $card_i$ be the cardinality of P_i. While $card_i$ is less than num_b, we add the next background observation of each point measurement. This results in an equal distribution of local ground points around the vehicle. We apply RANSAC [7] to compute a local approximation of the ground surface using P_i. Each plane is defined by a corresponding normal vector $\mathbf{n}_i$ and an arbitrary point $\mathbf{p}_i$ lying on the plane.

2.3 Scale Estimation using Environment Structure Constraints

In section 2.3, we exploit priors of object motion to improve the robustness of the reconstructed object trajectory. We assume that the object of interest moves on a locally planar surface. In this case the distance of each object point $\mathbf{o}_{j,i}^{(b)}$ to the ground is constant for all cameras i. The reconstructed trajectory shows this property only for the true scale ratio and non-degenerated camera motion. For example, a degenerate case occurs when the camera moves exactly parallel to a planar object motion. For a more detailed discussion of degenerated camera motions see [18].

Scale Ratio Estimation using a Single View Pair We use the term *view* to denote cameras and corresponding local ground planes. The signed distance of an object point $\mathbf{o}_{j,i}^{(b)}$ to the ground plane can be computed according to $d_{j,i} = \mathbf{n}_i \cdot (\mathbf{o}_{j,i}^{(b)} - \mathbf{p}_i)$, where $\mathbf{p_i}$ is an arbitrary point on the local ground plane and $\mathbf{n_i}$ is the corresponding normal vector. If the object moves on top of the approximated terrain ground the distance $d_{j,i}$ is be independent of a specific camera i. Thus, for a specific point and different cameras the relation shown in equation (3) holds.

$$\mathbf{n}_i \cdot (\mathbf{o}_{j,i}^{(b)} - \mathbf{p}_i) = \mathbf{n}_{i'} \cdot (\mathbf{o}_{j,i'}^{(b)} - \mathbf{p}_{i'}). \tag{3}$$

Substituting equation (1) in equation (3) results in (4)

$$\mathbf{n}_i \cdot (\mathbf{c}_i^{(b)} + r \cdot \mathbf{v}_{j,i}^{(b)} - \mathbf{p}_i) = \mathbf{n}_{i'} \cdot (\mathbf{c}_{i'}^{(b)} + r \cdot \mathbf{v}_{j,i'}^{(b)} - \mathbf{p}_{i'}) \tag{4}$$

Solving equation (4) for r yields equation (5)

$$r = \frac{\mathbf{n}_{i'} \cdot (\mathbf{c}_{i'}^{(b)} - \mathbf{p}_{i'}) - \mathbf{n}_i \cdot (\mathbf{c}_i^{(b)} - \mathbf{p}_i)}{(\mathbf{n}_i \cdot \mathbf{v}_{j,i}^{(b)} - \mathbf{n}_{i'} \cdot \mathbf{v}_{j,i'}^{(b)})}. \tag{5}$$

Equation (5) allows us to determine the scale ratio r between object and background reconstruction using the extrinsic parameters of two cameras and corresponding ground approximations.

Scale Ratio Estimation using View Pair Ranking The accuracy of the estimated scale ratio r in equation (5) is subject to the condition of the parameters of the particular view pair. For instance, if the numerator or denominator is close to zero, small errors in the camera poses or ground approximations may result in negative scale ratios. In addition, wrongly estimated local plane normal vectors may disturb camera-plane distances. We tackle these problems by combining two different view pair rankings. The first ranking uses for each view pair the difference of the camera-plane distances, i.e. $|\mathbf{n}_{i'} \cdot (\mathbf{c}_{i'}^{(b)} - \mathbf{p}_{i'}) - \mathbf{n}_i \cdot (\mathbf{c}_i^{(b)} - \mathbf{p}_i)|$. The second ranking reflects the quality of the local ground approximation w.r.t. the object reconstruction. A single view pair allows to determine $\left|\mathcal{P}^{(o)}\right|$ different scale ratios. For a view pair with stable camera registrations and well reconstructed local planes the variance of the corresponding scale ratios is small. This allows us to determine ill conditioned view pairs. The second ranking uses the scale ratio difference to order the view pairs. We sort the view pairs by weighting both ranks equally.
This ranking is crucial to deal with motion trajectories close to degenerated cases. In contrast to other methods, this ranking allows to estimate consistent vehicle motion trajectories, even if the majority of local ground planes are badly reconstructed. Concretely, this approach allows to determine a consistent trajectory using a single suitable view pair.
Let vp denote the view pair with the lowest overall rank. The final scale ratio is determined by using a least squares method w.r.t. all equations of vp according to equation (6). Let i and i' denote the image indices corresponding to vp.

$$\begin{bmatrix} \dots \\ \mathbf{n}_i \cdot \mathbf{v}_{j,i}^{(b)} - \mathbf{n}_{i'} \cdot \mathbf{v}_{j,i'}^{(b)} \\ \dots \\ \mathbf{n}_i \cdot \mathbf{v}_{j+1,i}^{(b)} - \mathbf{n}_{i'} \cdot \mathbf{v}_{j+1,i'}^{(b)} \\ \dots \end{bmatrix} \cdot r = \begin{bmatrix} \dots \\ \mathbf{n}_{i'}(\mathbf{c}_{i'}^{(b)} - \mathbf{p}_{i'}) - \mathbf{n}_i \cdot (\mathbf{c}_i^{(b)} - \mathbf{p}_i) \\ \dots \\ \mathbf{n}_{i'}(\mathbf{c}_{i'}^{(b)} - \mathbf{p}_{i'}) - \mathbf{n}_i \cdot (\mathbf{c}_i^{(b)} - \mathbf{p}_i) \\ \dots \end{bmatrix} \tag{6}$$

2.4 Scale Estimation Baseline using Intersection Constraints

The baseline is motivated by the fact, that some of the reconstructed points at the bottom of a vehicle should lie in the proximity of the ground surface of the environment. Consider for example 3D points triangulated at the wheels of a vehicle. This approach works only if at least one camera-object-point-ray intersects the local ground surface approximations. For each camera we use equation (2) to generate a set of direction vectors $\mathbf{v}_{j,i}^{(b)}$. For non-orthogonal direction vectors $\mathbf{v}_{j,i}^{(b)}$ and normal vectors $\mathbf{n}_i$ we compute the ray-plane-intersection parameter for each camera-object-point-pair according to equation (7)

$$r_{j,i} = (\mathbf{p}_i - \mathbf{c}_i^{(b)}) \cdot \mathbf{n}_i \cdot (\mathbf{v}_{j,i}^{(b)} \cdot \mathbf{n}_i)^{-1}. \tag{7}$$

Let r_i denote the smallest ray-plane-intersection parameter of image i. This parameter corresponds to a point at the bottom of the vehicle lying on the planar

approximation of the ground surface. Substituting r in equation (1) with r_i results in a vehicle point cloud being on top of the local terrain approximation corresponding to image i. Thus, r_i represents a value close to the scale ratio of object and background reconstruction. To increase the robustness of the computed scale ratio, we use the median r of all image specific scale ratios r_i to determine the final scale ratio.

$$r = \text{median}(\{\min(\{r_{j,i} \mid j \in \{1, \dots, |\mathcal{P}^{(o)}|\}\}) \mid i \in \mathcal{I}\}), \tag{8}$$

Here, $\mathcal{I}$ denotes the set of images indices. Cameras without valid intersection parameter r_i are not considered for the computation of r.

3 Virtual Object Motion Trajectory Dataset

To quantitatively evaluate the quality of the reconstructed object motion trajectory we require accurate object and environment models as well as object and camera poses at each time step. The simultaneous capturing of corresponding ground truth data with sufficient quality is difficult to achieve. For example, one could capture the environment geometry with LIDAR sensors and the camera / object pose with an additional system. However, the registration and synchronization of all these different modalities is a complex and cumbersome process. The result will contain noise and other artifacts like drift. To tackle these issues we exploit virtual models. Previously published virtually generated and virtually augmented datasets, like [20,21,8,28], provide data for different application domains and do not include three-dimensional ground truth information. We build a virtual world including an urban environment, animated vehicles as well as predefined vehicle and camera motion trajectories. This allows us to compute spatial and temporal error free ground truth data. We exploit procedural generation of textures to avoid artificial repetitions. Thus, our dataset is suitable for evaluating SfM algorithms.

3.1 Trajectory Dataset

We use the previously created virtual world to build a new vehicle trajectory dataset. The dataset consists of 35 sequences capturing five vehicles in different urban scenes. Fig. 2 shows some example images. The virtual video sequences cover a high variety of vehicle and camera poses. The vehicle trajectories reflect common vehicle motions include vehicle acceleration, different curve types and motion on changing slopes. We use the path-tracing render engine Cycles [1] to achieve photo realistic rendering results. We observed that the removal of artificial path-tracing artifacts using denoising is crucial to avoid degenerated SfM reconstructions.

The dataset includes 6D vehicle and camera poses for each frame as well as ground truth meshes of corresponding vehicle models. In contrast to measured ground truth data, virtual ground truth data is free of noise and shows no spatial registration or temporal synchronization inaccuracies. The dataset contains

Fig. 2: Frames from sequences contained in the presented virtual vehicle trajectory dataset.

semantic segmentations of vehicles, ground and background to separate the reconstruction task from specific semantic segmentation and tracking approaches. In addition to the virtual data, the dataset also includes the computed reconstruction results. We will make our evaluation scripts publicly available to foster future analysis of vehicle trajectory estimation.

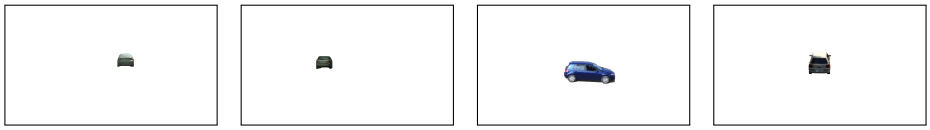
3.2 Virtual World

We used Blender [1] to create a virtual world consisting of a city surrounded by a countryside. We exploit procedural generation to compute textures of large surfaces, like streets and sidewalks, to avoid degenerated Structure from Motion results caused by artificial texture repetitions. The virtual world includes different assets like trees, traffic lights, streetlights, phone booths, bus stops and benches. We collected a set of publicly available vehicle assets to populate the scenes. We used skeletal animation, also referred to as rigging, for vehicle animation. This includes wheel rotation and steering w.r.t. the motion trajectory as well as consistent vehicle placement on uneven ground surfaces. The animation of wheels is important to avoid unrealistic wheel point triangulations. We adjusted the scale of vehicles and virtual environment using Blender's unit system. This allows us to set the virtual space in relation to the real world. The extent of the generated virtual world corresponds to one square kilometer. We exploit environment mapping to achieve realistic illumination. With Blender's built-in tools, we defined a set of camera and object motion trajectories. This allows us to determine the exact 3D pose of cameras and vehicles at each time step.

4 Experiments and Evaluation

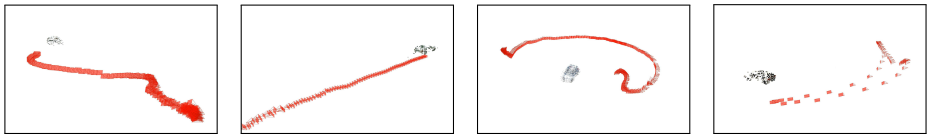
Fig. 3 shows qualitative results using driving sequences from the Cityscapes dataset [4] as well as real and virtual drone footage. For sequences with multi-

(a) Input Frame.

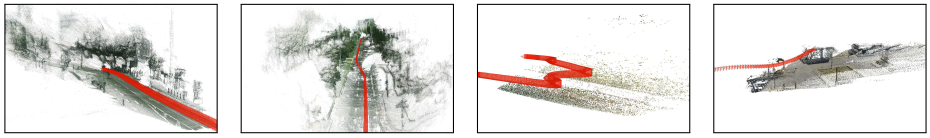


(b) Object Segmentation.

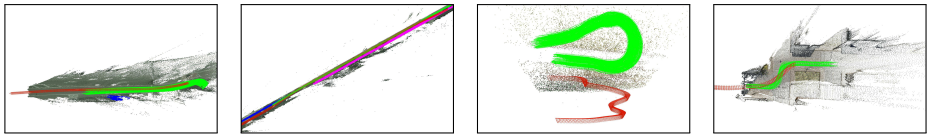
(c) Background Segmentation.



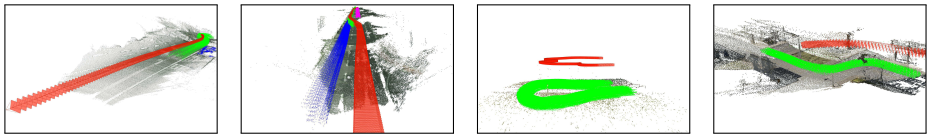
(d) Object Reconstruction.



(e) Background Reconstruction.

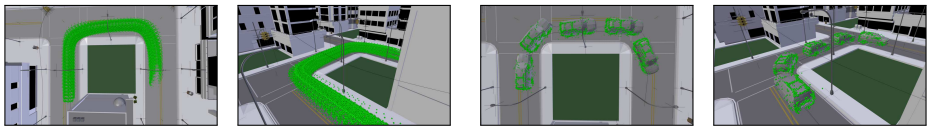


(f) Trajectory Reconstruction (Top View).



(g) Trajectory Reconstruction (Side View).

Fig. 3: Vehicle trajectory reconstruction using two sequences (first two columns) from the Cityscape dataset [4], one sequence captured by a drone (third column) as well as one virtually generated sequence of our dataset (last column). Object segmentations and object reconstructions are shown for one of the vehicles visible in the scene. The reconstructed cameras are shown in red. The vehicle trajectories are colored green, blue and pink.



(a) Example of a registered vehicle trajectory in the ground truth coordinate frame system.

(b) Example of a vehicle trajectory with the corresponding ground truth vehicle model at selected frames.

Fig. 4: Vehicle trajectory registration for quantitative evaluation.

ple vehicle instances only one vehicle segmentation and reconstruction is shown. However, the trajectory reconstruction results contain multiple reconstructed vehicle trajectories. Fig. 4 depicts the quantitative evaluation using our dataset. Fig. 4a shows the object point cloud transformed into the virtual world coordinate frame system. The vehicle motion trajectory has been registered with the virtual environment using the approach described in section 4.2. Fig. 4b shows the overlay of transformed points and the corresponding virtual ground truth vehicle model.
To segment the two-dimensional vehicle shapes, we follow the approach presented in [2]. In contrast to [2], we used [15] and [11] to segment and track visible objects, respectively. We considered the following SfM pipelines for vehicle and background reconstructions: Colmap [23], OpenMVG [16], Theia [27] and VisualSfM [30]. Our vehicle trajectory reconstruction pipeline uses Colmap for vehicle and OpenMVG for background reconstructions, since Colmap and OpenMVG created in our experiments the most reliable vehicle and background reconstructions. We enhanced the background point cloud using [22].

4.1 Quantitative Vehicle Trajectory Evaluation

We use the dataset presented in section 3 to quantitatively evaluate the proposed vehicle motion trajectory reconstruction approach. The evaluation is based on vehicle, background and ground segmentations included in the dataset. This allows us to show results independent from the performance of specific instance segmentation and tracking approaches. We compare the proposed method with the baseline presented in section 2.4 using 35 sequences contained in the dataset. We automatically register the reconstructed vehicle trajectory to the ground truth using the method described in section 4.2. We compute the shortest distance of each vehicle trajectory point to the vehicle mesh in ground truth coordinates. For each sequence we define the trajectory error as the average trajectory-point-mesh distance. Fig. 5 shows for each sequence the trajectory error in meter. The average trajectory error per vehicle using the full dataset is shown in table 1. Overall, we achieve a trajectory error of 0.31 meter. The error of the vehicle trajectory reconstructions reflects four types of computational inaccuracies: deviations of camera poses w.r.t. vehicle and background point clouds, wrong triangulated vehicle points as well as scale ratio discrepancies. Fig. 5 compares

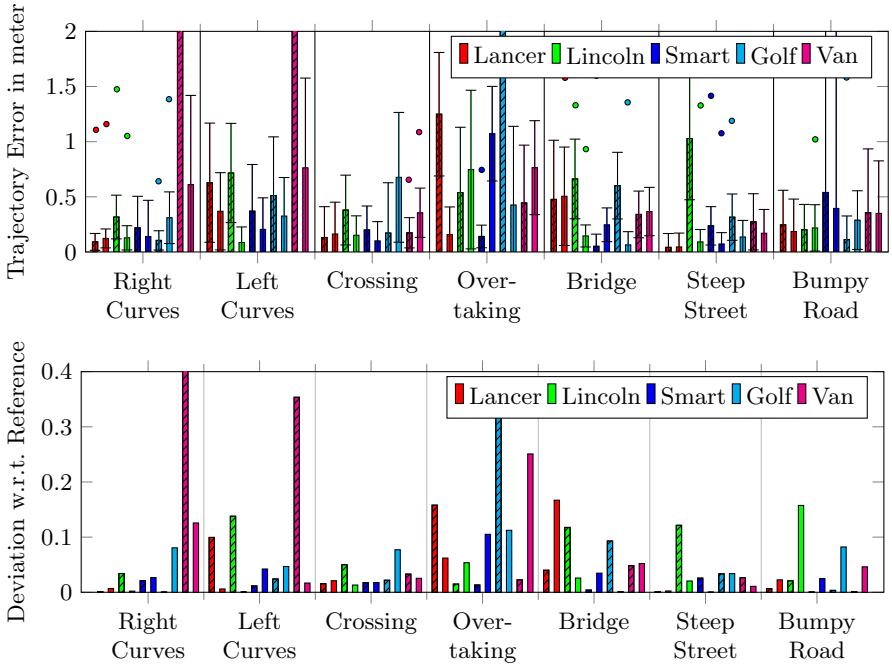


Fig. 5: Quantitative evaluation of the trajectory reconstruction computed by our proposed method (plain colored bars) and the *baseline* (dashed bars). We evaluate seven different vehicle trajectories (*Right Curves*, ...) and five different vehicle models (*Lancer*, ...). The top figure shows the trajectory error in meter, which reflects deviations of camera poses w.r.t. vehicle and background point clouds, wrong triangulated vehicle points as well as scale ratio discrepancies. The circles show the trajectory error of the most distant points. The intervals denote the standard deviation of the trajectory errors. The reference scale ratios used in the bottom figure are only subject to the registration of the background reconstruction and the virtual environment. The figure is best viewed in color.

the estimated scale ratios of the proposed and the baseline method w.r.t. the reference scale ratio. The reference scale ratio computation is described in section 4.3. The overall estimated scale ratio deviation w.r.t. the reference scale per vehicle is shown in table 1. The provided reference scale ratios are subject to the registration described in section 4.2. Wrongly reconstructed background camera poses may influence the reference scale ratio. The *van* vehicle reconstruction was only partial successful on the sequences *crossing*, *overtaking* and *steep street*. The SfM algorithm registered 19%, 60% and 98% of the images, respectively. The vehicle reconstruction of the *smart* model contained 74% of the *crossing* input vehicle images. Here, we use the subset of registered images to perform the evaluation. The camera and the vehicle motion in *bumpy road* sim-

Scale Ratio	Average Scale Ratio Deviation					Average Trajectory Error [m]				
Est. Type	Lancer	Lincoln	Smart	Golf	Van	Lancer	Lincoln	Smart	Golf	Van
Baseline	0.05	0.07	**0.01**	0.08	0.13	0.42	0.53	**0.25**	0.95	1.68
Ours	**0.04**	**0.04**	0.04	**0.06**	**0.08**	**0.20**	**0.23**	0.33	**0.33**	**0.47**

Table 1: Summary of the conducted evaluation. The second column shows the deviation of the estimated scale ratio w.r.t to the reference scale ratio. The third column contains the average distances of the full dataset in meter. Overall, the trajectory error of the baseline and our approach is 0.77m and 0.31m.

ulate a sequence close to a degenerated case, i.e. equation (5) is ill conditioned for all view pairs.

4.2 Registration of Background Reconstruction and Virtual Environment

A common approach to register different coordinate systems is to exploit 3D-3D correspondences. To determine points in the virtual environment corresponding to background reconstruction points one could create a set of rays from each camera center to all visible reconstructed background points. The corresponding environment points are defined by the intersection of these rays with the mesh of the virtual environment. Due to the complexity of our environment model this computation is in terms of memory and computational effort quite expensive. Instead, we use the algorithm presented in [29] to estimate a similarity transformation $\mathbf{T}_s$ between the cameras contained in the background reconstruction and the virtual cameras used to render the corresponding video sequence. This allows us to perform 3D-3D-registrations of background reconstructions and the virtual environment as well as to quantitatively evaluate the quality of the reconstructed object motion trajectory. We use the camera centers as input for [29] to compute an initial reconstruction-to-virtual-environment transformation. Depending on the shape of the camera trajectory there may be multiple valid similarity transformations using camera center positions. In order to find the semantically correct solution we enhance the original point set with camera pose information, i.e. we add points reflecting up vectors $\mathbf{u}_i^{(b)} = {\mathbf{R}_i^{(b)}}^T \cdot (0,1,0)^T$ and forward vectors $\mathbf{f}_i^{(b)} = {\mathbf{R}_i^{(b)}}^T \cdot (0,0,1)^T$. For the reconstructed cameras, we adjust the magnitude of these vectors using the scale computed during the initial similarity transformation. We add the corresponding end points of up $\mathbf{c}_i^{(b)} + m \cdot \mathbf{u}_i^{(b)}$ as well as viewing vectors $\mathbf{c}_i^{(b)} + m \cdot \mathbf{f}_i^{(b)}$ to the camera center point set. Here, m denotes the corresponding magnitude.

4.3 Reference Scale Ratio Computation

As explained in section 4.1 the presented average trajectory errors in Fig. 5 are subject to four different error sources. To evaluate the quality of the scale

ratio estimation between object and background reconstruction we provide corresponding reference scale ratios. The scale ratios between object reconstruction, background reconstruction and virtual environment are linked via the relation $r_{(ov)} = r_{(ob)} \cdot r_{(bv)}$, where $r_{(ov)}$ and $r_{(bv)}$ are the scale ratios between object and background reconstructions and virtual environment, respectively. The scale ratios $r_{(ob)}$ in Fig. 5 express the spatial relation of vehicle and background reconstructions. The similarity transformation $\mathbf{T}_s$ defined in section 4.2 implicitly contains information about the scale ratio $r_{(bv)}$ between background reconstruction and virtual environment. To compute $r_{(ov)}$ we use corresponding pairs of object reconstruction and virtual cameras. We use the extrinsic parameters of the object reconstruction camera to transform all 3D points in the object reconstruction into camera coordinates. Similarly, the object mesh with the pose of the corresponding frame is transformed into the camera coordinates leveraging the extrinsic camera parameters of the corresponding virtual camera. The ground truth pose and shape of the object mesh is part of the dataset. In camera coordinates we generate rays from the camera center (i.e. the origin) to each 3D point $\mathbf{o}_j^{(i)}$ in the object reconstruction. We determine the shortest intersection $\mathbf{m}_j^{(i)}$ of each ray with the object mesh in camera coordinates. This allows us to compute the reference scale ratio $r_{(ov)}^{ref}$ according to equation (9) and the reference scale ratio $r_{(ob)}^{ref}$ according to $r_{(ob)}^{ref} = r_{(ov)}^{ref} \cdot {r_{(bv)}}^{-1}$.

$$r_{(ov)}^{ref} = \text{med}(\{\text{med}(\{\|\mathbf{m}_j^{(i)}\| \cdot \|\mathbf{o}_j^{(i)}\|^{-1} | j \in \{1, \ldots, n_J\}\} | i \in \{1, \ldots, n_I\}\}). \quad (9)$$

The reference scale ratio $r_{(ob)}^{ref}$ depends on the quality of the estimated camera poses in the background reconstruction, i.e. $r_{(bv)}$, and may slightly differ from the true scale ratio.

5 Conclusions

This paper presents a pipeline to reconstruct the three-dimensional trajectory of vehicles using monocular video data. We propose a novel constraint to estimate consistent object motion trajectories and demonstrate the effectiveness of our approach showing vehicle trajectory reconstructions using drone footage and driving sequences from the Cityscapes dataset. Due to the lack of 3D object motion trajectory benchmark datasets with suitable ground truth data, we present a new virtual dataset to quantitatively evaluate object motion trajectories. The dataset contains rendered videos of urban environments and accurate ground truth data including semantic segmentations, object meshes as well as object and camera poses for each frame. The proposed algorithm achieves an average reconstruction-to-ground-truth distance of 0.31 m evaluating 35 trajectories. In future work, we will analyze the performance of the proposed pipeline in more detail with focus on minimal object sizes, object occlusions and degeneracy cases. In addition, we intend to integrate previously published scale estimation approaches. These will serve together with our dataset as benchmark references for future vehicle/object motion trajectory reconstruction algorithms.

References

1. Blender Online Community: Blender - a 3d modelling and rendering package (2016), http://www.blender.org 8, 9
2. Bullinger, S., Bodensteiner, C., Arens, M.: Instance flow based online multiple object tracking. In: IEEE International Conference on Image Processing (ICIP). IEEE (2017) 3, 11
3. Chhaya, F., Reddy, N.D., Upadhyay, S., Chari, V., Zia, M.Z., Krishna, K.M.: Monocular reconstruction of vehicles: Combining SLAM with shape priors. In: IEEE International Conference on Robotics and Automation (ICRA). IEEE (2016) 2, 5
4. Cordts, M., Omran, M., Ramos, S., Rehfeld, T., Enzweiler, M., Benenson, R., Franke, U., Roth, S., Schiele, B.: The cityscapes dataset for semantic urban scene understanding. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR). IEEE (2016) 9, 10
5. Dai, J., He, K., Sun, J.: Instance-aware semantic segmentation via multi-task network cascades. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR). IEEE (2016) 2
6. Farabet, C., Couprie, C., Najman, L., LeCun, Y.: Learning hierarchical features for scene labeling. IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI) (2013) 2
7. Fischler, M.A., Bolles, R.C.: Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography. ACM Communications **24**(6) (1981) 6
8. Gaidon, A., Wang, Q., Cabon, Y., Vig, E.: Virtual worlds as proxy for multi-object tracking analysis. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR). IEEE (2016) 8
9. Hartley, R.I., Zisserman, A.: Multiple View Geometry in Computer Vision. Cambridge University Press, ISBN: 0521540518, second edn. (2004) 5
10. He, K., Gkioxari, G., Dollar, P., Girshick, R.: Mask r-cnn. In: IEEE International Conference on Computer Vision (ICCV). IEEE (2017) 2
11. Ilg, E., Mayer, N., Saikia, T., Keuper, M., Dosovitskiy, A., Brox, T.: Flownet 2.0: Evolution of optical flow estimation with deep networks. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR). IEEE (2017) 3, 11
12. Kundu, A., Krishna, K.M., Jawahar, C.V.: Realtime multibody visual slam with a smoothly moving monocular camera. In: IEEE International Conference on Computer Vision (ICCV). IEEE (2011) 2, 3
13. Lebeda, K., Hadfield, S., Bowden, R.: 2d or not 2d: Bridging the gap between tracking and structure from motion. In: Cremers, D., Reid, I., Saito, H., Yang, M.H. (eds.) Asian Conference on Computer Vision (ACCV). Springer International Publishing (2015) 2
14. Lee, B., Daniilidis, K., Lee, D.D.: Online self-supervised monocular visual odometry for ground vehicles. In: IEEE International Conference on Robotics and Automation (ICRA). IEEE (2015) 2, 5
15. Li, Y., Qi, H., Dai, J., Ji, X., Wei, Y.: Fully convolutional instance-aware semantic segmentation. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR). IEEE (2017) 2, 3, 11
16. Moulon, P., Monasse, P., Marlet, R., Others: Openmvg. an open multiple view geometry library. (2013) 2, 4, 11

17. Namdev, R.K., Krishna, K.M., Jawahar, C.V.: Multibody vslam with relative scale solution for curvilinear motion reconstruction. In: IEEE International Conference on Robotics and Automation (ICRA). IEEE (2013) 2
18. Ozden, K.E., Cornelis, K., Eycken, L.V., Gool, L.J.V.: Reconstructing 3d trajectories of independently moving objects using generic constraints. Computer Vision and Image Understanding **96**(3) (2004) 2, 3, 5, 6
19. Park, H.S., Shiratori, T., Matthews, I., Sheikh, Y.: 3d trajectory reconstruction under perspective projection. International Journal of Computer Vision **115**(2) (2015) 2, 3, 5
20. Richter, S.R., Vineet, V., Roth, S., Koltun, V.: Playing for data: Ground truth from computer games. In: Leibe, B., Matas, J., Sebe, N., Welling, M. (eds.) European Conference on Computer Vision (ECCV). LNCS, Springer International Publishing (2016) 8
21. Ros, G., Sellart, L., Materzynska, J., Vazquez, D., Lopez, A.M.: The synthia dataset: A large collection of synthetic images for semantic segmentation of urban scenes. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR). IEEE (2016) 8
22. Schönberger, J.L., Zheng, E., Frahm, J.M., Pollefeys, M.: Pixelwise view selection for unstructured multi-view stereo. In: Leibe, B., Matas, J., Sebe, N., Welling, M. (eds.) European Conference on Computer Vision (ECCV). Springer International Publishing (2016) 11
23. Schönberger, J.L., Frahm, J.M.: Structure-from-motion revisited. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR). IEEE (2016) 2, 4, 11
24. Shelhamer, E., Long, J., Darrell, T.: Fully convolutional networks for semantic segmentation. IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI) **39**(4) (2017) 2, 5
25. Snavely, N., Seitz, S.M., Szeliski, R.: Photo tourism: Exploring photo collections in 3d. ACM Transactions on Graphics **25**(3) (2006) 2
26. Song, S., Chandraker, M., Guest, C.C.: High accuracy monocular SFM and scale correction for autonomous driving. IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI) **38**(4) (2016) 2, 5
27. Sweeney, C.: Theia Multiview Geometry Library: Tutorial & Reference. University of California Santa Barbara. (2014) 2, 11
28. Tsirikoglou, A., Kronander, J., Wrenninge, M., Unger, J.: Procedural modeling and physically based rendering for synthetic data generation in automotive applications. CoRR (2017) 8
29. Umeyama, S.: Least-squares estimation of transformation parameters between two point patterns. IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI) **13**(4) (1991) 13
30. Wu, C.: Visualsfm: A visual structure from motion system (2011) 2, 11
31. Yuan, C., Medioni, G.G.: 3d reconstruction of background and objects moving on ground plane viewed from a moving camera. In: IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR). IEEE (2006) 2, 3, 5