Monocular 3D Vehicle Trajectory Reconstruction Using Terrain Shape Constraints

Sebastian Bullinger, Christoph Bodensteiner and Michael Arens
Fraunhofer Institute of Optronics, System Technologies and Image Exploitation, 76275 Ettlingen, Germany
Email: {sebastian.bullinger, christoph.bodensteiner, michael.arens}@iosb.fraunhofer.de

Abstract— **This work proposes a novel approach to reconstruct three-dimensional vehicle trajectories in monocular video sequences. We leverage state-of-the-art instance-aware semantic segmentation and optical flow methods to compute object video tracks on pixel level. This approach uses Structure from Motion to determine camera poses relative to vehicle instances and environment structures. We parameterize vehicle trajectories with a single variable by combining object and background reconstructions. The naive combination of vehicle and environment reconstruction results in inconsistent motion trajectories due to the scale ambiguity of SfM. We determine consistent object trajectories by projecting dense vehicle reconstructions on the terrain surface. Our scale ratio estimation approach shows no degenerated camera-vehicle-motions. We demonstrate the usefulness of our approach using publicly available video data of driving scenarios. We extend this evaluation showing trajectory reconstruction results using drone footage. We use synthetic data of vehicles in urban environments to evaluate the proposed algorithm. We achieve an average reconstruction-to-ground-truth distance of 0.17 meter.**

I. Introduction

A. Trajectory Reconstruction

The reconstruction of three-dimensional vehicle trajectories is crucial not only for autonomous vehicles, but also for driver assistance systems. Recent Structure from Motion (SfM) [1], [2] and Multi-View Stereo (MVS) [3] libraries are able to build city scale 3D environment models using monocular cameras. In contrast to other sensing modalities, like LIDAR, commodity cameras are cheaper, lighter and require less energy. While this is in general a desirable aspect for all types of transportation systems, it is of particular interest for small sensor platforms, like drones. We propose a method to reconstruct three-dimensional trajectories of vehicles using SfM and MVS techniques with a single camera.

Due to the scale ambiguity of SfM additional motion assumptions are required to compute three-dimensional vehicle trajectories consistent to terrain structures. [4]–[6] focus on driving scenarios where the relative pose and height of the camera is known. These approaches are not applicable to scenarios with variable camera poses like drones or motorcycles.

[7] presents a principle to reconstruct independent object motions exploiting non-accidentalness. By applying additional motion constraints this approach allows determining the scale ratio between vehicle and environment reconstruction. For instance, [8]–[10] tackle the scale ambiguity by proposing specific object-camera-motion constraints. In contrast to previous works, we propose a degeneracy-free method to compute object-background-scale-ratios by exploiting terrain geometry constraints.

To reconstruct moving vehicles in video data with Structure-from-Motion or Visual SLAM approaches it is necessary to separate object instances and background structures. The joint triangulation of independently moving vehicle and background feature correspondences results in missing object or scene points as well as omitted object or scene cameras. [11], [12] use motion segmentation and keypoint tracking to detect, track and reconstruct three-dimensional vehicle trajectories. Since motion segmentation fails to segment stationary objects and keypoint tracking is not suitable for occlusion handling, we apply instance-aware semantic segmentation and Structure-from-Motion techniques as basis for the proposed vehicle trajectory reconstruction pipeline. Our method does not rely on specific camera pose constraints such as fixed camera-ground-angles or known camera-ground-distances. We demonstrate the effectiveness of our approach showing results for a variety of input videos including driving scenarios and sequences captured by drones.

B. Contribution

(1) This paper presents a system for vehicle trajectory reconstruction in monocular video data. We exploit state-of-the-art Structure from Motion and semantic segmentation approaches to compute three-dimensional vehicle trajectories. (2) Structure from Motion reconstructions are scale ambiguous. Thus, the naive combination of object and background reconstruction results in inconsistent vehicle motion trajectories. We propose a novel constraint to compute object-background-scale-ratios and vehicle trajectories consistent to terrain shape and image observations. In contrast to previously published approaches, the presented constraint shows no degenerated motion cases. (3) The computation of the proposed constraint requires the intersection of camera-object-rays and terrain shape. We describe an efficient intersection computation approach exploiting depth buffer values of terrain mesh renderings of corresponding camera poses. (4) We present two filter steps to remove outliers in dense object reconstructions. (5) We perform a quantitative evaluation of our vehicle trajectory reconstruction algorithm using virtual data and (6) show the feasibility of the al-

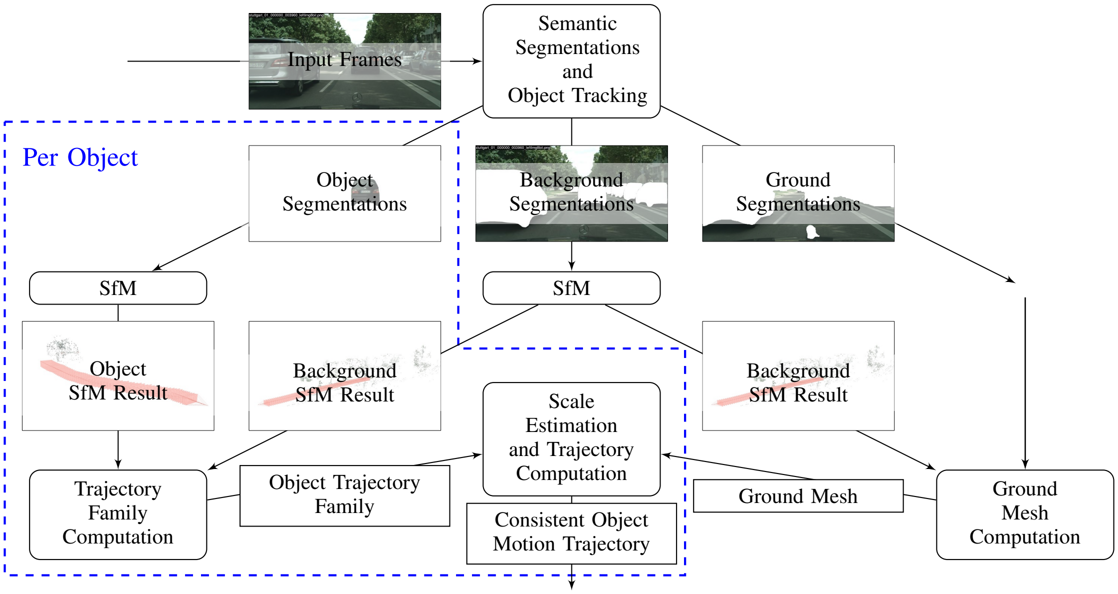


Fig. 1: Pipeline of the vehicle trajectory reconstruction approach. Computation results are represented by boxes with corners and computation steps by boxes with rounded corners. Arrows show computational dependencies.

gorithm using the CityScapes dataset and additional drone footage.

C. Related Work

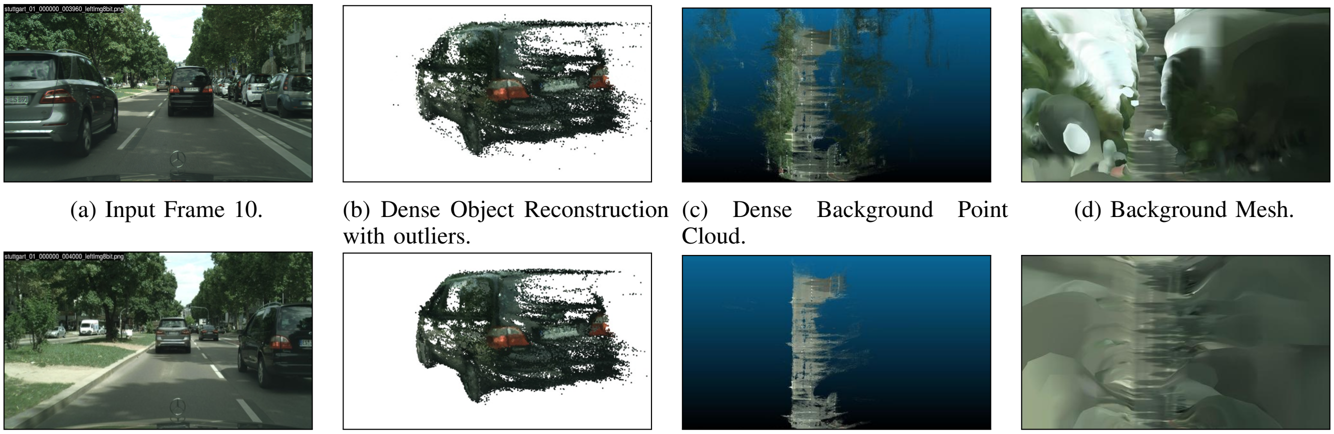
The presented vehicle trajectory reconstruction approach builds on top of instance-aware semantic segmentation and Structure from Motion methods. While early ConvNets designed for semantic segmentation applied patchwise training [13], current Fully Convolutional Networks [14] are trained end-to-end. Recent works [15]–[17] perform semantic segmentation on instance level, i.e. pixels are associated with instance identifiers in addition to class labels. Currently, there are two popular categories of rigid SfM approaches: sequential and global SfM. Sequential SfM methods [1], [2], [18]–[20] register images iteratively whereas global SfM approaches [1], [20] compute camera poses jointly, for example, by rotation averaging.
The naive combination of object and environment reconstruction leads to inconsistent motion trajectories due to the scale ambiguity of Structure-from-Motion results. [7] introduces the reconstruction of 3D trajectories of independently moving objects exploiting non-accidental motion constraints. [8] leverages the same principle while assuming that the object of interest moves perpendicular to a single ground plane. [11] applies a bearing-only-tracker based multibody VSLAM to reconstruct trajectories of moving vehicles. [21] approximates the ground with locally planar surfaces and assumes that distances of object points to corresponding ground representations are constant. [9] represents the motion of single 3D points with linear combinations of trajectory basis vectors. This approach is suitable to reconstruct independently moving point sets. In contrast to previously published three-dimensional trajectory reconstruction approaches, our method does not show degenerated motion cases. [21] presents a virtual dataset to quantitatively evaluate vehicle trajectories using monocular video data. [7]–[9], [11] show only qualitative results.

II. Object Motion Trajectory Reconstruction

Fig. 1 outlines the pipeline of our approach. The input is an ordered image sequence. We track two-dimensional object shapes on pixel level across video sequences following the scheme proposed in [22]. In contrast to [22], we used [17] for instance-aware semantic segmentation and [23] for optical flow computations to increase the robustness of the tracking pipeline. We use the term *object images* to denote images that show only pixels of single vehicle instances. In contrast, *background images* depict exclusively environment structures. We apply SfM [1], [2] to object and background images as shown in Fig. 1. The corresponding Structure from Motion results are denoted with $sfm^{(o)}$ and $sfm^{(b)}$. We use $sfm^{(o)}$ and $sfm^{(b)}$ to define three-dimensional scale-dependent object motion trajectories. We project dense vehicle point clouds onto reconstructed terrain meshes to determine consistent vehicle environment scale ratios.

A. Object Trajectory Representation

Previous works [7], [8], [11], [21] propose different notations to describe scale dependent object trajectory representations. We provide a brief description of scale dependent vehicle trajectories following the notation presented in [21]. Without loss of generality, we describe scale dependent trajectory representations for single vehicles. [21] denotes the reconstructed points in $sfm^{(o)}$ and $sfm^{(b)}$ with $\mathbf{o}_j^{(o)} \in \mathcal{P}^{(o)}$ and $\mathbf{b}_k^{(b)} \in \mathcal{P}^{(b)}$. The indices j and k identify specific points in the corresponding point cloud. The superscripts (o) and (b) distinguish between elements (e.g. vectors and camera poses) defined w.r.t. the coordinate frame systems of $sfm^{(o)}$


stuttgart_01_000000_003960_leftImg8bit.png
stuttgart_01_000000_004000_leftImg8bit.png

(a) Input Frame 10.

(b) Dense Object Reconstruction with outliers.

(c) Dense Background Point Cloud.

(d) Background Mesh.

(e) Input Frame 50.

(f) Dense Object Reconstruction without outliers.

(g) Dense Ground Point Cloud.

(h) Ground Mesh.

Fig. 2: Intermediate results for scale ratio computation. Results are computed using the Cityscape dataset [24].

and $sfm^{(b)}$.

We use the term *reconstructed camera* to denote extrinsic and intrinsic parameters of registered object and background images. We associate reconstructed cameras in $sfm^{(o)}$ with their counterpart in $sfm^{(b)}$ and vice versa, i.e. reconstructed cameras belonging to the same input frame. Reconstructed cameras without corresponding counterparts are removed from the reconstructions.

Let $\mathbf{R}_i^{(o)}$ and $\mathbf{c}_i^{(o)}$ denote camera rotation and center of input image i contained in the object reconstruction $sfm^{(o)}$. The rotation and center are defined w.r.t. the reconstructed object point cloud $\mathbf{o}_j^{(o)} \in \mathcal{P}^{(o)}$. Equation (1) allows us to convert vehicle points $\mathbf{o}_j^{(o)}$ given in object coordinates to vehicle points $\mathbf{o}_j^{(i)}$ given in camera coordinates of camera i.

$$\mathbf{o}_j^{(i)} = \mathbf{R}_i^{(o)} \cdot (\mathbf{o}_j^{(o)} - \mathbf{c}_i^{(o)}) \tag{1}$$

A pair of reconstructed cameras in $sfm^{(o)}$ and $sfm^{(b)}$ share the same camera coordinate system. The camera center $\mathbf{c}_i^{(b)}$ and rotation $\mathbf{R}_i^{(b)}$ contained in the background reconstruction $sfm^{(b)}$ allows us to convert object points $\mathbf{o}_j^{(i)}$ in camera coordinates to object points $\mathbf{o}_{j,i}^{(b)}$ in background coordinates as shown in equation (2)

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

Due to the scale ambiguity of SfM the naive combination of vehicle and environment reconstruction results in inconsistent motion trajectories [25]. We model the scale ambiguity by extending equation (2) with an additional variable representing the scale ratio r between object and background reconstruction. Equation (3) allows to convert vehicle points $\mathbf{o}_j^{(o)}$ defined in object coordinates to vehicle points $\mathbf{o}_{j,i}^{(b)}$ in environment coordinates of camera i.

$$\begin{aligned}\mathbf{o}_{j,i}^{(b)} &= \mathbf{c}_i^{(b)} + r \cdot \mathbf{R}_i^{(b)^T} \cdot \mathbf{R}_i^{(o)} \cdot (\mathbf{o}_j^{(o)} - \mathbf{c}_i^{(o)}) \\ &:= \mathbf{c}_i^{(b)} + r \cdot \mathbf{v}_{j,i}^{(b)}\end{aligned} \tag{3}$$

To define a scale ratio parameterized object motion trajectory we represent the object points for each frame with equation (3). Substituting r in equation (3) with the real scale ratio allows us to compute a consistent vehicle motion trajectory.

B. Scale Ratio Estimation using Shape Constraints

We tackle the problem of determining consistent object-background-scale-ratios by exploiting geometric consistency constraints applicable to ground restricted object categories, like vehicles. In contrast to previous works, our approach does neither rely on restrictions of camera and object motions nor specific camera poses. The proposed scale-ratio estimation approach shows no degenerated cases, in which a consistent object trajectory computation is impossible. Our method exploits the fact that some vehicle points should touch the terrain surface, like 3D points corresponding to the wheels of a car.

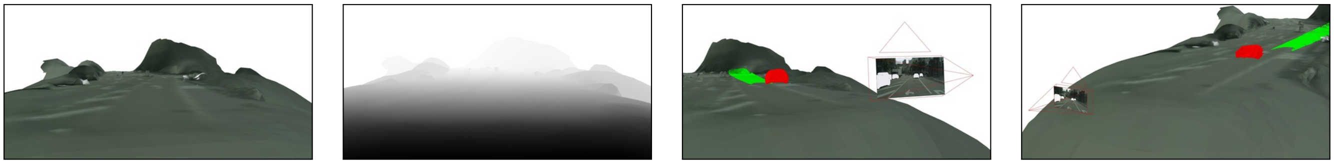
To ensure the presence of suitable 3D points we enhance the points in $sfm^{(o)}$ by leveraging the Multi-View Stereo (MVS) algorithm presented in [3]. In contrast to sparse SfM algorithms, the MVS library [3] reliably triangulates points at wheels of driving vehicles. We exploit the previously computed instance-aware object segmentations to determine outliers in the dense object point cloud. Let $\mathbf{p}_{j,i}$ be the homogeneous image projection of a point $\mathbf{o}_j^{(o)}$ given in object world coordinates w.r.t. to camera i.

$$\mathbf{p}_{j,i} = \mathbf{K}_i \mathbf{R}_i^{(o)} (\mathbf{o}_j^{(o)} - \mathbf{c}_i^{(o)}) \tag{4}$$

$\mathbf{K}_i$ denotes the corresponding calibration matrix. We define the object affinity of a 3D point $\mathbf{o}_j^{(o)}$ in the dense object point cloud according to equation (5)

$$o_j = \sum_i \theta_i(\mathbf{p}_{j,i}) \;/\; \sum_i \sigma_i(\mathbf{p}_{j,i}). \tag{5}$$

The pixel classification function $\theta_i(\mathbf{p}) = 1$, if $\mathbf{p}$ corresponds to the object in image i and $\theta_i(\mathbf{p}) = 0$, otherwise. $\sigma_i(\mathbf{p})$ takes the visibility into account, i.e. $\sigma_i(\mathbf{p}) = 1$, if $\mathbf{p}$ is visible in image i and $\sigma_i(\mathbf{p}) = 0$, otherwise. We classify an object



(a) Rendered Ground Mesh.

(b) Depth Buffer.

(c) Object Point Cloud and Corresponding Projected Points.

(d) Object Point Cloud and Corresponding Projected Points.

Fig. 3: Projection of the object point cloud (red) onto the ground mesh using the depth buffer. The projected points are shown in green. The inconsistent initial scale ratio becomes apparent by examining the distance between object points and corresponding projections. Results are computed using the Stuttgart01 sequence in the Cityscape dataset [24].

point j as outlier, if $o_j < 0.9$. This threshold is empirically determined and takes the robustness of the instance-aware segmentation computed with [17] into account. We apply statistical outlier removal to the previously computed object points using the standard deviation of the mean distance as outlier criterion. The mean distance is computed considering the five next neighbors. Fig. 2(b) and 2(f) show a dense object reconstruction with and without outliers, respectively.

We apply the MVS algorithm presented in [3] to the sparse background reconstruction $sfm^{(b)}$ to compute a dense background representation. We exploit ground segmentations to classify 3D points in the background point cloud as ground or non-ground points. Let $\mu(j)$ be the set of image indices used to triangulate background point j and $\boldsymbol{\nu}_{j,i}$ be the pixel position of the corresponding observation in image i. We define the ground affinity according to equation (6)

$$g_j = \frac{1}{|\mu(j)|} \sum_{i \in \mu(j)} \phi_i(\boldsymbol{\nu}_{j,i}). \tag{6}$$

The pixel classification function $\phi_i(\boldsymbol{\nu}) = 1$, if $\boldsymbol{\nu}$ corresponds to ground in image i and $\theta_i(\boldsymbol{\nu}) = 0$, otherwise. The (non)-ground segmentation is computed using [14]. We use the points $\mathbf{b}_j^{(b)}$ in the dense background point cloud with $g_j > 0.5$ to compute a dense ground point cloud. Fig. 2(c) shows the dense background reconstruction and Fig. 2(g) only the points classified as ground.

We use the algorithm described in [26] to compute watertight ground meshes. This allows us to inter- and extrapolate ground surface areas occluded by moving objects. We determine connected components in the ground mesh and remove isolated mesh parts. Fig. 2(h) shows an example of a computed ground mesh. The removal of non-ground points before computing the mesh speeds up the computation and leads to a more precise representation of the ground geometry.

To determine a consistent object-background-reconstruction scale ratio we use equation (3) to create for each camera i a set of vectors $\mathbf{v}_{j,i}^{(b)}$ pointing from the camera center $\mathbf{c}_i^{(b)}$ to the position $\mathbf{o}_{j,i}^{(b)}$ of point j. Let F denote the set of faces contained in the ground mesh and $h_{j,i}$ the ray defined by $\mathbf{c}_i^{(b)}$ and $\mathbf{v}_{j,i}^{(b)}$.

A naive approach to determine the closest ray-ground-mesh-intersection of a ray $h_{j,i}$ requires the ray-face-intersection and corresponding ray-face-intersection parameter computation for each face $f \in F$. This includes intersection tests with occluded faces and faces not visible in the field of view of the current background camera i. This makes the object-ground-ray intersection computation for all rays $h_{j,i}$ computationally expensive.

Instead of computing object-ground-ray intersections, we use the visualization toolkit (VTK) [27] to render the ground mesh from the perspective of camera i. We exploit the information stored in the depth buffer to determine 3D-3D object-ground correspondences. We determine for each point $\mathbf{o}_j^{(o)}$ the corresponding point $\mathbf{o}_j^{(i)} = \mathbf{R}_i^{(o)}(\mathbf{o}_j^{(o)} - \mathbf{c}_i^{(o)})$ in the camera coordinate system of camera i as well as the corresponding image projection $x_{i,j} = \mathbf{K}_i \mathbf{o}_j^{(i)}$. For $x_{i,j}$ we use the corresponding depth buffer value to determine a point $\mathbf{p}_j^{(i)}$ lying on the ground mesh surface with the same projection than $\mathbf{o}_j^{(i)}$ w.r.t. to camera i. We apply bilinear interpolation while accessing depth buffer values.

To determine a consistent scale ratio, we must find the smallest r, which satisfies $\|\mathbf{o}_j^{(i)}\| = r \cdot \|\mathbf{p}_j^{(i)}\|$ for an arbitrary point $\mathbf{o}_j^{(i)}$ in the object point cloud. We compute r according to equation (7) for each image i, separately.

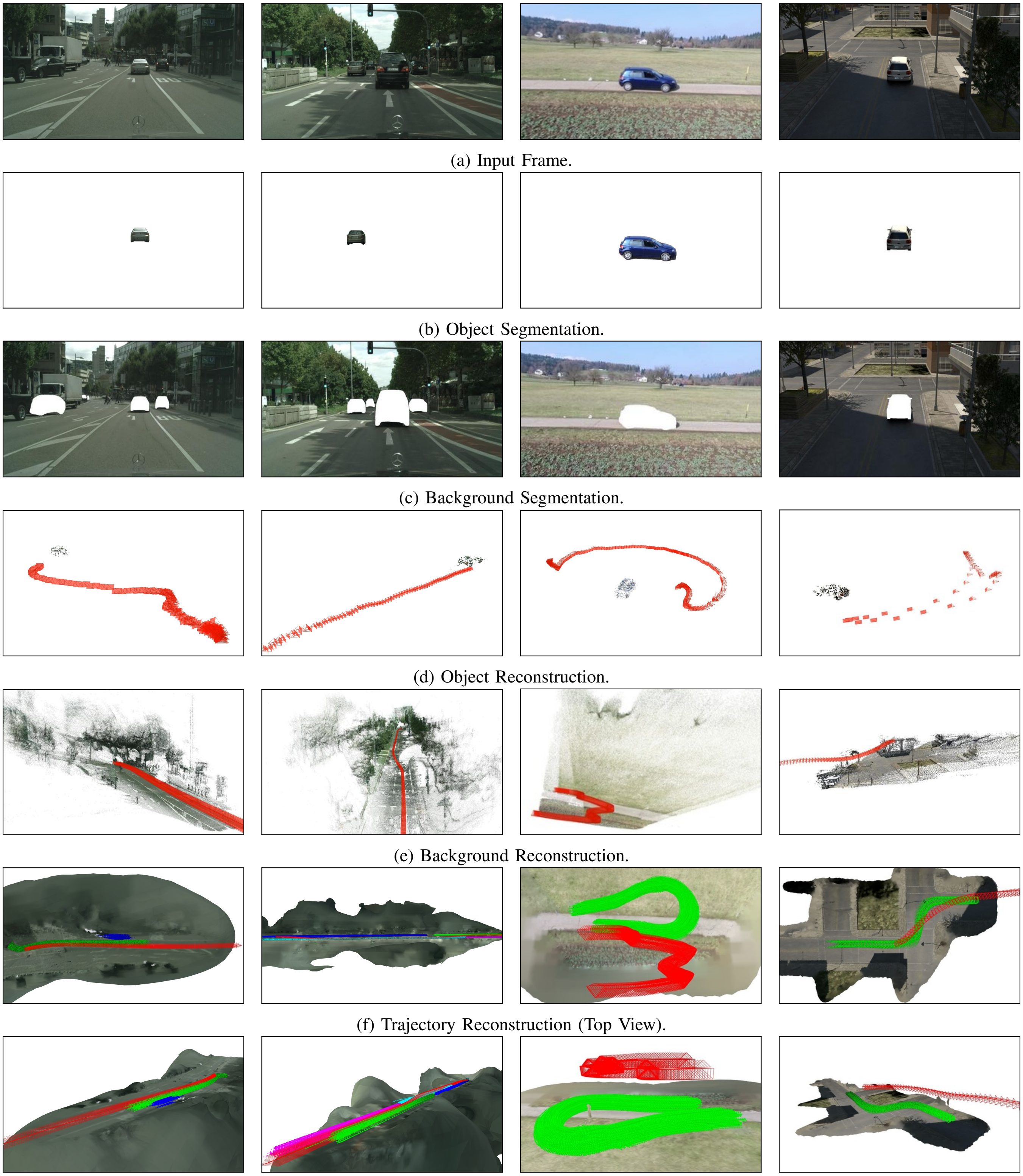
$$r_i = \min(\{\|\mathbf{p}_j^{(i)}\| \cdot (\|\mathbf{o}_j^{(i)}\|)^{-1} | j \in \{1, \ldots, |\mathcal{P}^{(o)}|\}\}) \tag{7}$$

The scale ratio and intersection parameter r_i corresponds to the point being closest to the ground surface, i.e. a point at the bottom of the vehicle. Plugging r_i in equation (3) for camera i places the object point cloud on top of the ground surface. Thus, the smallest ray-plane-intersection-parameter r_i represents the object-to-background-scale-ratio. We reconstruct the three-dimensional vehicle trajectory as defined in equation (8).

$$r = med(\{r_i | i \in \{1, \ldots, n_I\}\}) \tag{8}$$

Here, med denotes the median and n_I the number of images. We do not consider invalid image scale ratios r_i, i.e. cameras which have no camera-object-point-rays intersecting the ground representation.

To compute the final object trajectory we compute equation (3) for each point j at all time steps i. The removal of outliers greatly improves object trajectory visualizations, since a single outlier in the object reconstruction results in multiple outliers in the final object trajectory.



(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. 4: Qualitative evaluation of the proposed trajectory reconstruction approach. First and second column: driving sequences of the Cityscape dataset [24]. Third column: drone sequence. Fourth column: synthetic drone sequence of the dataset presented in [21]. Reconstructed cameras are shown in red. The last two rows show the reconstructed environment mesh with vehicle trajectories in green, blue, pink and teal. The figure is best viewed in color.

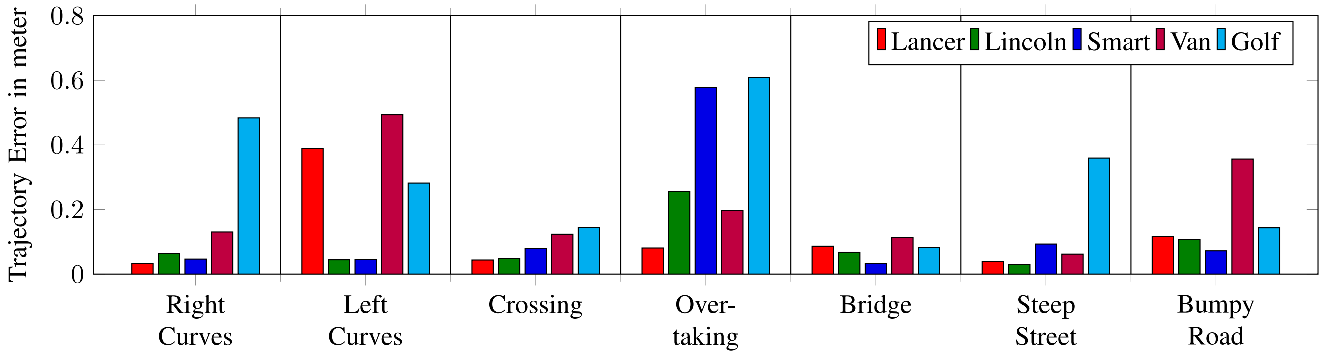
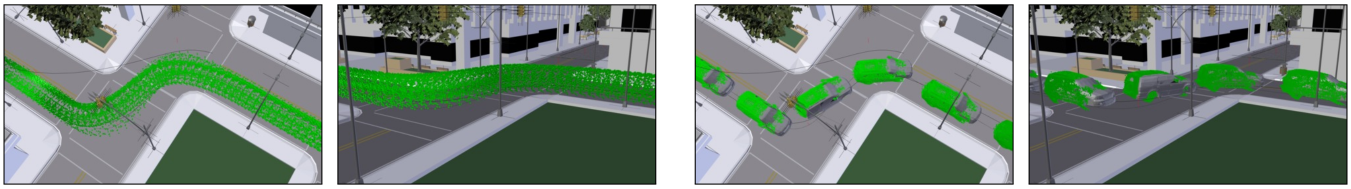


Fig. 5: Quantitative evaluation of the proposed method using the dataset presented in [21]. The dataset contains seven different vehicle trajectories (*Right Curves*, *Left Curves*, *Crossing* ...) and five different vehicle models (*Lancer*, *Lincoln Navigator*, ...). The figure shows the trajectory error in meter, which is the average trajectory-point-mesh distance, i.e. the shortest distance of each vehicle trajectory point to the vehicle mesh at the corresponding time step. The trajectory error is affected by object and background camera poses registration errors, incorrect vehicle point triangulations and scale ratio inaccuracies.



(a) Reconstructed vehicle trajectory in the coordinate frame system of the virtual environment.

(b) Registered vehicle trajectory at selected frames with corresponding ground truth vehicle models.

Fig. 6: Registration of the reconstructed vehicle trajectory for quantitative evaluation. The trajectory points are shown in green. The virtual environment is part of the synthetic dataset presented in [21]. The figure is best viewed in color.

III. Evaluation

Fig. 4 shows qualitative results of our vehicle trajectory reconstruction approach using publicly available video data [24]. We used the instance-aware segmentation in [17] and the optical flow algorithm in [23] to segment and track visible objects following the approach described in [22]. We observe that [23] outperforms [28] for large object displacements. We evaluated different SfM pipelines for object and environment reconstructions: Colmap [2], OpenMVG [1], Theia [20] and VisualSfM [19]. We notice that Colmap computes the best vehicle and OpenMVG the most stable environment reconstructions. We also applied different MVS algorithms for ground reconstruction: [29], MVE [30] and Colmap [3]. Using drone imagery all MVS approaches achieved decent results, however in driving scenarios only Colmap reconstructed the ground surface with satisfying quality.

Fig. 2(b) and Fig. 2(f) show the MVS object reconstruction result before and after the outlier removal described in section II-B. Fig. 2(c) shows the MVS background reconstruction and Fig. 2(h) the corresponding ground mesh.

Due to the lack of datasets with TV video data of driving vehicles and suitable 3D ground truth data, we show quantitative results using the synthetic dataset presented in [21]. The dataset consists of 35 sequences containing five vehicles and seven different motion paths of drone footage in an urban environment. The dataset provides pose and shape information of vehicles for each frame as ground truth. The dataset contains also ground truth camera poses used to render the sequences. The ground truth camera poses and the camera poses of the background reconstruction allow registering the reconstructed vehicle trajectory with the ground truth coordinate system. Fig. 6 shows an example. See [21] for more details about the registration of the trajectories with the ground truth coordinate system. Fig. 5 shows the trajectory error of all vehicles for each motion path. [21] defines the trajectory error as the average trajectory-point-mesh-distance, i.e. the shortest distance of each vehicle trajectory point to the vehicle mesh at the corresponding time step. In a few cases, the algorithms presented in [2] and [3] compute degenerated object point clouds, which decreases the quality of the corresponding scale ratio estimation. Table I shows the trajectory errors per vehicle in comparison to [21]. Our method achieves a trajectory error of 0.17 meter and outperforms the results reported in [21]. Table I highlights the importance of the outlier removal described in section II-B.

IV. Conclusions

This paper presents a pipeline to reconstruct the three-dimensional trajectory of vehicles using monocular video data. We leverage Multi-View Stereo techniques to compute

Scale Ratio	Average Trajectory Error (meter)				
Est. Type	Lancer	Lincoln	Smart	Van	Golf
[21]	0.20	0.23	0.33	0.33	0.47
Ours	**0.11**	**0.09**	**0.14**	**0.21**	**0.30**
Ours (no o.r.)	1.13	1.51	1.40	1.29	1.47

TABLE I: Trajectory error per vehicle of the benchmark dataset presented in [21]. Our approach achieves an average trajectory error of 0.17 m considering all sequences and outperforms the method presented in [21] by 0.14 m. The last row shows the results of our method without outlier removal (o.r.).

accurate vehicle and environment models. We propose a novel approach to estimate consistent vehicle trajectories using terrain shape constraints. In contrast to previous approaches, the presented scale ratio constraint shows no degenerated motion cases. We show the effectiveness of the proposed vehicle trajectory reconstruction approach using drone footage and publicly available video data of driving scenarios. We perform a quantitative evaluation of the proposed approach using synthetic drone footage. We achieve an average trajectory error of 0.17 m evaluating 35 different sequences. In future work, we will analyze breakdown points of the proposed pipeline in more detail. This includes minimal object sizes and object occlusions.

References

[1] P. Moulon, P. Monasse, R. Marlet, and Others, “Openmvg. an open multiple view geometry library.” 2013.

[2] J. L. Schönberger and J.-M. Frahm, “Structure-from-motion revisited,” in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[3] J. L. Schönberger, E. Zheng, M. Pollefeys, and J.-M. Frahm, “Pixelwise view selection for unstructured multi-view stereo,” in *European Conference on Computer Vision (ECCV)*, 2016.

[4] S. Song, M. Chandraker, and C. C. Guest, “High accuracy monocular SFM and scale correction for autonomous driving,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 38, no. 4, pp. 730–743, 2016. [Online]. Available: http://dx.doi.org/10.1109/TPAMI.2015.2469274

[5] B. Lee, K. Daniilidis, and D. D. Lee, “Online self-supervised monocular visual odometry for ground vehicles,” in *2015 IEEE International Conference on Robotics and Automation (ICRA)*, May 2015, pp. 5232–5238.

[6] F. Chhaya, N. D. Reddy, S. Upadhyay, V. Chari, M. Z. Zia, and K. M. Krishna, “Monocular reconstruction of vehicles: Combining SLAM with shape priors,” in *2016 IEEE International Conference on Robotics and Automation, ICRA 2016, Stockholm, Sweden, May 16-21, 2016*, 2016, pp. 5758–5765. [Online]. Available: http://dx.doi.org/10.1109/ICRA.2016.7487799

[7] K. E. Ozden, K. Cornelis, L. V. Eycken, and L. J. V. Gool, “Reconstructing 3d trajectories of independently moving objects using generic constraints,” *Computer Vision and Image Understanding*, vol. 96, no. 3, pp. 453–471, 2004. [Online]. Available: http://dx.doi.org/10.1016/j.cviu.2004.03.015

[8] C. Yuan and G. G. Medioni, “3d reconstruction of background and objects moving on ground plane viewed from a moving camera,” in *2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 2006), 17-22 June 2006, New York, NY, USA*, 2006, pp. 2261–2268. [Online]. Available: http://dx.doi.org/10.1109/CVPR.2006.16

[9] H. S. Park, T. Shiratori, I. Matthews, and Y. Sheikh, “3d trajectory reconstruction under perspective projection.” *International Journal of Computer Vision*, vol. 115, no. 2, pp. 115–135, 2015.

[10] R. K. Namdev, K. M. Krishna, and C. V. Jawahar, “Multibody vslam with relative scale solution for curvilinear motion reconstruction,” in *2013 IEEE International Conference on Robotics and Automation*, May 2013, pp. 5732–5739.

[11] A. Kundu, K. M. Krishna, and C. V. Jawahar, “Realtime multibody visual slam with a smoothly moving monocular camera,” in *ICCV*, 2011.

[12] K. Lebeda, S. Hadfield, and R. Bowden, “2D or not 2D: Bridging the gap between tracking and structure from motion,” in *Proceedings, Asian Conference on Computer Vision (ACCV)*, 2014.

[13] C. Farabet, C. Couprie, L. Najman, and Y. LeCun, “Learning hierarchical features for scene labeling,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, August 2013.

[14] E. Shelhamer, J. Long, and T. Darrell, “Fully convolutional networks for semantic segmentation,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 39, no. 4, pp. 640–651, 2017. [Online]. Available: http://dx.doi.org/10.1109/TPAMI.2016.2572683

[15] J. Dai, K. He, and J. Sun, “Instance-aware semantic segmentation via multi-task network cascades,” in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[16] Y. Li, H. Qi, J. Dai, X. Ji, and Y. Wei, “Fully convolutional instance-aware semantic segmentation,” in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017.

[17] K. He, G. Gkioxari, P. Dollar, and R. Girshick, “Mask r-cnn,” in *The IEEE International Conference on Computer Vision (ICCV)*, Oct 2017.

[18] N. Snavely, S. M. Seitz, and R. Szeliski, “Photo tourism: Exploring photo collections in 3d,” *ACM Trans. Graph.*, vol. 25, no. 3, pp. 835–846, July 2006. [Online]. Available: http://doi.acm.org/10.1145/1141911.1141964

[19] C. Wu, “Visualsfm: A visual structure from motion system,” 2011.

[20] C. Sweeney, *Theia Multiview Geometry Library: Tutorial & Reference*, University of California Santa Barbara., 2014.

[21] S. Bullinger, C. Bodensteiner, M. Arens, and R. Stiefelhagen, “3d vehicle trajectory reconstruction in monocular video data using environment structure constraints,” in *IEEE European Conference on Computer Vision (ECCV)*, 2018, to appear.

[22] S. Bullinger, C. Bodensteiner, and M. Arens, “Instance flow based online multiple object tracking,” in *IEEE International Conference on Image Processing (ICIP)*, 2017.

[23] Y. Hu, R. Song, and Y. Li, “Efficient coarse-to-fine patchmatch for large displacement optical flow,” in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016.

[24] M. Cordts, M. Omran, S. Ramos, T. Rehfeld, M. Enzweiler, R. Benenson, U. Franke, S. Roth, and B. Schiele, “The cityscapes dataset for semantic urban scene understanding,” in *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[25] R. I. Hartley and A. Zisserman, *Multiple View Geometry in Computer Vision*, 2nd ed. Cambridge University Press, ISBN: 0521540518, 2004.

[26] M. Kazhdan and H. Hoppe, “Screened poisson surface reconstruction,” *ACM Trans. Graph.*, vol. 32, no. 3, pp. 29:1–29:13, July 2013. [Online]. Available: http://doi.acm.org/10.1145/2487228.2487237

[27] W. Schroeder, K. Martin, and B. Lorensen, *The Visualization Toolkit*, 4th ed. Kitware, 2006.

[28] E. Ilg, N. Mayer, T. Saikia, M. Keuper, A. Dosovitskiy, and T. Brox, “Flownet 2.0: Evolution of optical flow estimation with deep networks,” in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017. [Online]. Available: http://lmb.informatik.uni-freiburg.de/Publications/2017/IMKDB17

[29] C. Barnes, E. Shechtman, A. Finkelstein, and D. B. Goldman, “PatchMatch: A randomized correspondence algorithm for structural image editing,” *ACM Transactions on Graphics (Proc. SIGGRAPH)*, vol. 28, no. 3, Aug. 2009.

[30] S. Fuhrmann, F. Langguth, N. Moehrle, M. Waechter, and M. Goesele, “MVE – An Image-Based Reconstruction Environment,” *Computer and Graphics*, 2015.