16th International Conference on Machine Vision Applications (MVA)
National Olympics Memorial Youth Center, Tokyo, Japan, May 27-31, 2019.

3D Object Trajectory Reconstruction using Stereo Matching and Instance Flow based Multiple Object Tracking

Sebastian Bullinger, Christoph Bodensteiner and Michael Arens
Fraunhofer IOSB
76275 Ettlingen, Germany
<firstname>.<lastname>@iosb.fraunhofer.de

Abstract

This paper presents a method to reconstruct three-dimensional object motion trajectories in stereo video sequences. We apply stereo matching to each image pair of a stereo sequence to compute corresponding binocular disparities. By combining instance-aware semantic segmentation techniques and optical flow cues, we track two-dimensional object shapes on pixel level. This allows us to determine for each frame pair object specific disparities and corresponding object points. By applying Structure from Motion (SfM) we compute camera poses with respect to background structures. We embed the vehicle trajectories into the environment reconstruction by combining the object point cloud of each image pair with corresponding camera poses contained in the background SfM reconstruction. We show qualitative results on the KITTI and CityScapes dataset and compare our method quantitatively with previously published monocular approaches on synthetic data of vehicles in an urban environment. We achieve an average trajectory error of 0.11 *meter.*

1 Introduction

1.1 Trajectory Reconstruction

The perception of three-dimensional object trajectories is crucial for many application domains, such as autonomous driving and augmented reality. Stereo matching [1] is a widely used approach to infer 3D scene information provided by stereo cameras. Stereo matching exploits rectified stereo image pairs to determine pixel correspondences along so called scan lines. Using pixel disparities and stereo camera parameters allows us to triangulate dense scene points for each frame. Since stereo matching uses only image information of stereo image pairs to triangulate object points, this technique is not able to distinguish between static and dynamic points in the scene. To tackle this issue our pipeline leverages recent advances in instance-aware semantic segmentation to track two-dimensional object shapes and corresponding disparities on pixel level. This allows us to triangulate object specific points using stereo matching.
Since our approach uses stereo matching instead Structure from Motion for object reconstruction the method is not hampered by incorrectly registered camera poses caused by small object sizes, reflecting surfaces and changing illumination. In addition, object point clouds derived by stereo matching show usually higher point densities than Visual Slam or Structure from Motion reconstruction results.
We use Structure from Motion to determine camera poses relative to the environment for each time step. This allows us to embed the stereo matching based triangulated object points in a common coordinate frame system and to compute three-dimensional object motion trajectories.

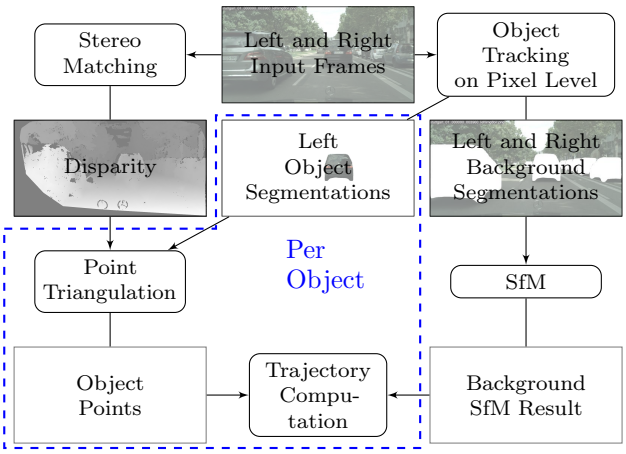


Figure 1: Overview of the Trajectory Reconstruction Pipeline. Boxes with corners denote computation results and boxes with rounded corners denote computation steps.

1.2 Related Work

[3, 4, 5] are three widely used off-the-shelf stereo matching methods. Recently, [6] and [7] presented two ConvNet based stereo matching approaches outperforming previous state-of-the-art on the Stereo Robust Vision Challenge [8]. The usage of these methods is limited, since the corresponding ConvNet models require input data matching the image ratios used for training and validation.
The authors in [9] compute stereo matching and class segmentation jointly using Conditional Random Fields.

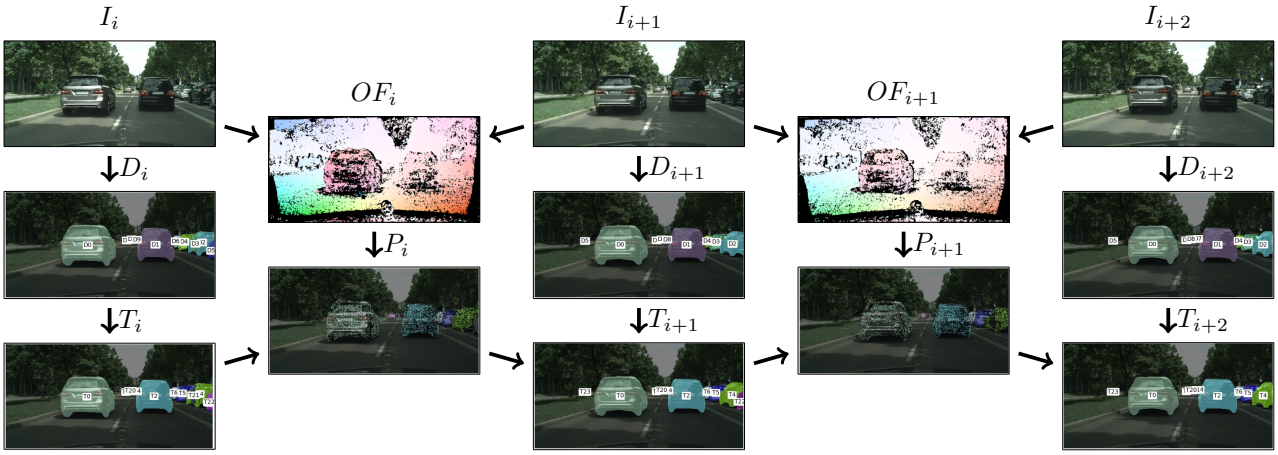


Figure 2: Multiple Object Tracking Scheme. The variables have the following meaning. I: image, OF: optical flow, D: detection, P: Prediction, T: Tracker State, i: image index. Arrows show the relation of computation steps. A computation step depends on the results connected with incoming arrows. The optical flow color coding used is defined in [2]. The figure is best viewed in color.

Furthermore, [10] leverages Markov Random Fields to perform joint optimization of stereo matching and class-agnostic object segmentation. In contrast, our method allows to compute stereo matching results associated with instance information as well as class labels.
Recently, several works determined object models including object shape and pose using stereo matching based point triangulations. [11] combines 2D object bounding box detections and 3D stereo depth measurements. Thus, also background structures are considered as object points. The detections and measurements are tracked with a 2D-3D Kalman filter to compute three-dimensional bounding box proposals for each object. [12] leverages a deformable vehicle shape prior to reconstruct 3D pose and shape. [13] tracks objects in 3D using [14] and imposes a common shape and a motion model by combining the information acquired by multiple frames corresponding to the same track.

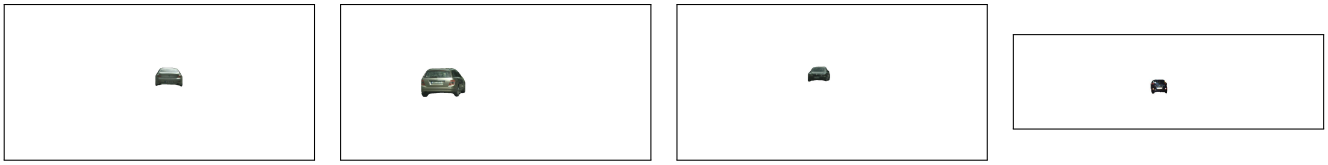
1.3 Contribution

The core contributions of this work are as follows: (1) We present a new pipeline to reconstruct three-dimensional trajectories of moving objects using stereo video data. (2) Earlier methods used bounding box detections to create stereo matching based object reconstructions. Our approach determines binocular disparities on pixel level, which avoids the triangulation of environment points during object reconstruction. (3) In contrast to previous joint stereo matching and segmentation methods, our approach allows to determine a class label as well as a corresponding object video identifier for each triangulated object point. (4) Due to the lack of suitable real-world benchmark datasets, we demonstrate the effectiveness of our method using synthetic data of vehicles in an urban environment.

2 Trajectory Reconstruction

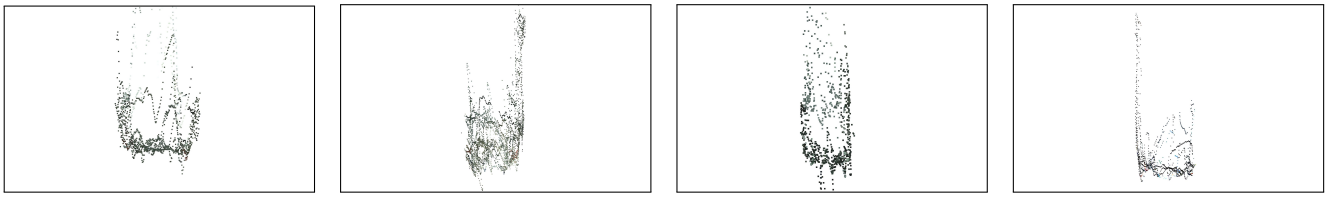
Fig. 1 shows an overview of the proposed object trajectory reconstruction pipeline. We perform stereo matching to obtain pixel disparity values for each image pair of the stereo camera. Following the Multiple Object Tracking (MOT) approach presented in [17] we track objects on pixel level in the images captured by the left sensor of the stereo camera. We use background images as input for Structure from Motion [18] to compute an environment model and associated stereo camera poses. For each object we leverage corresponding segmentation masks to determine object specific disparity values and to triangulate object points. Embedding the object points into the environment reconstruction for each time step allows us to determine the three-dimensional object motion trajectory.
In the following, i denotes the image index and the time step, respectively. The used MOT scheme is outlined in Fig. 2. We leverage the instance-aware semantic segmentation approach presented in [19] to compute object detections D_i on pixel level in all left images. The optical flow cues OF_i [20] allow us to predict two-dimensional object shapes into subsequent images. Let P_i denote the corresponding predictions. We associate the predictions P_i and detections D_{i+1} to determine the tracker state T_{i+1} at time $i+1$.
We use the complement of instance-aware semantic segmentations [19] to determine environment images. Applying SfM [18] to left and right background images enables us to compute stereo camera poses. Let $\mathbf{R}_i \in SO(3)$ and $\mathbf{c}_i \in \mathbb{R}^3$ denote the corresponding camera rotations and centers, respectively. The usage of the background images of the right camera for SfM allows to increase the robustness and removes the scale ambiguity

(a) Left Input Frame.

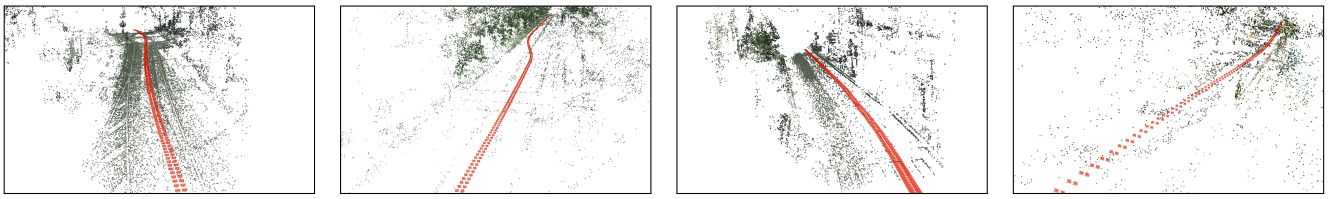


(b) Left Object Segmentation.

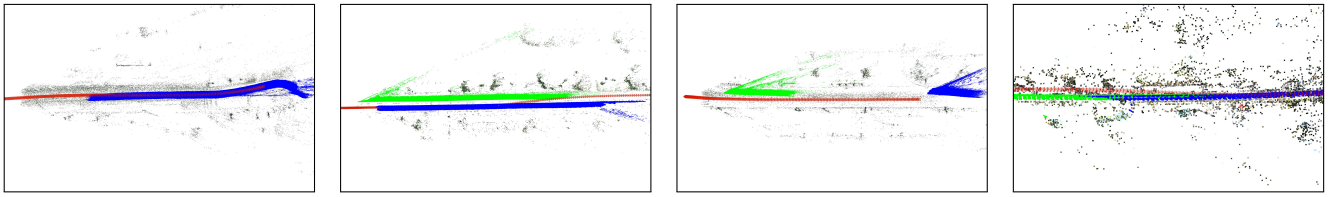
(c) Left Background Segmentation.



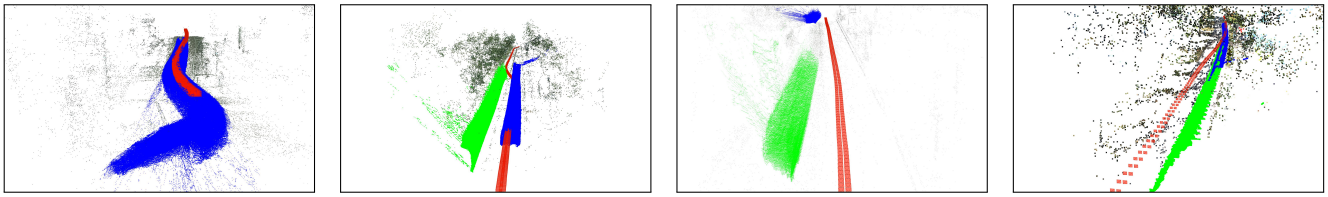
(d) Object Point Cloud (Top View).



(e) Background Structure from Motion Reconstruction.



(f) Trajectory Reconstruction (Top View).



(g) Trajectory Reconstruction (Side View).

Figure 3: Object trajectory reconstruction using three stereo sequences (stuttgart01-stuttgart03) included in the Cityscapes dataset [15] and one stereo sequence (2011_09_26_drive_0013) of the KITTI dataset [16]. Object segmentations and reconstructions are shown for one of the vehicles visible in the scene. The reconstructed camera poses are shown in red. The vehicle trajectories are colored in green and blue. The figure is best viewed in color.

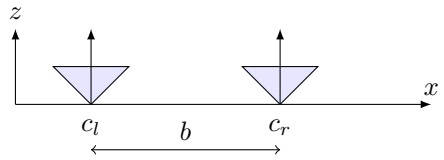


Figure 4: Coordinate frame system of the stereo camera. c_l and c_r denote the centers of the left and right camera and b denotes the corresponding baseline. x and z are the coordinate axis of the stereo camera system.

of the computed reconstruction results.
Stereo matching [3, 4] based point triangulation exploits the relative poses of the left and the right sensor of a stereo camera to determine three-dimensional scene points. Corresponding matches are determined along so called scan lines and allow to define pixelwise disparity functions $d_i(\cdot)$ for each time step.
Without loss of generality, we describe the trajectory reconstruction for a single object. In the following, $(u, v) \in \mathcal{P}_i$ denotes the set of pixels representing the current object in image i. Fig. 4 shows the setup of the stereo camera system and the corresponding coordinate frame systems, i.e. the x axis is pointing to the right, the y axis downwards and the z axis forward. We use the disparity-to-depth mapping matrix $\mathbf{Q}$ according to equation (1) to determine homogeneous points $(x_u, y_v, z, w_{u,v,i})$ corresponding to the pixel disparity triplets $(u, v, d_i(u, v))$ of the left image.

$$\begin{bmatrix} x_u \\ y_v \\ z \\ w_{u,v,i} \end{bmatrix} = \underbrace{\begin{bmatrix} 1 & 0 & 0 & -c_u \\ 0 & 1 & 0 & -c_v \\ 0 & 0 & 0 & f \\ 0 & 0 & \frac{-1}{b} & \frac{c_u - c'_u}{b} \end{bmatrix}}_{\mathbf{Q}} \cdot \begin{bmatrix} u \\ v \\ d_i(u,v) \\ 1 \end{bmatrix} \quad (1)$$

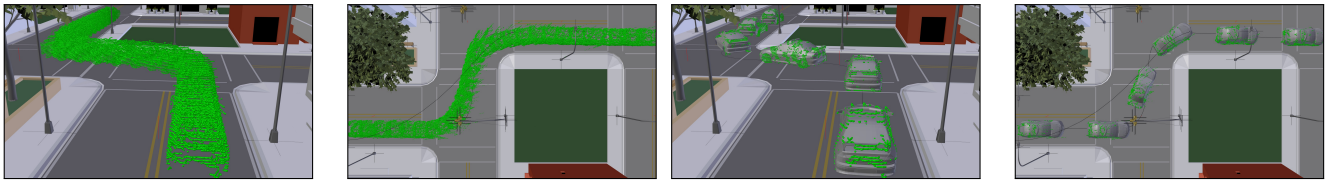
Here, (c_u, c_v) and f denote the principal point and the focal length in pixels. b is the extent of the stereo camera baseline in the background SfM coordinate frame system. This ensures, that object points and camera poses are correctly scaled. Normalizing $(x_u, y_v, z, w_{u,v,i})^T$ yields the actual three-dimensional object point $\mathbf{o}_{u,v,i} = (\frac{x_u}{w_{u,v,i}}, \frac{y_v}{w_{u,v,i}}, \frac{z}{w_{u,v,i}})^T$ in camera coordinates. We decrease computation time and memory consumption using only every second object pixel for triangulation.
We observe that incorrectly estimated disparity values lead to distant, isolated object points - usually close to the object boundary. We assume that each object consists of a single connected component, i.e. each object point has neighbor points with similar depth values. Equation (2) shows the depth error δz of triangulated points using stereo matching. For more details see [21].

$$\delta z = \frac{z^2 \cdot \delta d}{b \cdot f} \quad (2)$$

Here, b, f and δd are the corresponding stereo camera baseline, focal length and disparity deviation values. Equation (2) shows a) the estimated depth error δz increases quadratic with the corresponding distance z and b) the estimation of close points is more reliable than the computation of distant points. Defining a threshold for disparity variation between adjacent pixels allows us to compute dynamic depth intervals of valid object points, which take the corresponding depth value into account.
For each object pixel $(u, v) \in \mathcal{P}_i$, we consider a local $l \times l$ neighborhood of object points $\mathcal{N} = \{\mathbf{o}_{u+m,v+n,i} \mid m, n \in \{-\lfloor \frac{l}{2} \rfloor, \ldots, \lfloor \frac{l}{2} \rfloor\} \wedge (u+m, v+n) \in \mathcal{P}_i\}$ around (u, v). Let $z_{u,v,i}$ denote the depth value corresponding to $\mathbf{o}_{u,b,i}$. We consider $\mathbf{o}_{u,v,i}$ as outlier, if there is a point $\mathbf{o}_{u+m,v+n,i} \in \mathcal{N}$ with $z_{u,v,i} > z_{u+m,v+n,i} + \delta z_{u+m,v+n,i}$. In this case $\mathbf{o}_{u+m,v+n,i}$ lies closer to the camera and according to equation (1) the corresponding depth can be estimated more reliably.
To compute the full object trajectory we transform the object point cloud for each time step i into world coordinates with $\mathbf{p}_{j,i} = \mathbf{c}_i + \mathbf{R}_i{}^T \cdot \mathbf{o}_{j,i}$.

3 Experiments

We considered [3, 4, 5] for stereo matching of object points and [23, 18, 24] to reconstruct environment structures and corresponding camera poses. Evaluations on the KITTI and the CityScapes dataset demonstrated that a) [3] produces frequently vertical disparity artifacts, which result in incorrectly triangulated object points and b) [4] provides a better trade-off between reconstruction quality and computation time than [3, 5]. Also, we observe that [18] outperforms [23, 24] w.r.t. the number of correctly registered cameras.
In our experiments we used the following parameters for outlier removal: $l = 5$ to define the local 5×5 neighborhood areas and $\delta d = 5$ pixels to constrain the valid triangulation depth. For instance, this results in a depth range of $\delta Z = 0.6$ and $\delta Z = 1.3$ meter using the parameters of the KITTI dataset and a triangulation depth of $Z = 10$ and $Z = 15$ meter, respectively.
Fig. 3 shows a qualitative evaluation of the proposed 3D object trajectory reconstruction approach on sequences of the CityScapes and the KITTI dataset. The Cityscapes and the KITTI dataset are captured with a stereo camera baseline of 0.22 m and 0.54 m, respectively. According to the Fresnel equations, reflections at surfaces increase while the viewing angle between camera and surface decreases. This effect cause huge sets of incorrect disparity values in specific images, for instance, when vehicles are overtaking.
Due to the lack of suitable real-world benchmark datasets, we used the synthetic vehicle trajectory dataset presented in [22] to quantitatively evaluate the proposed pipeline. The dataset contains 35 sequences



(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.

Figure 5: Registration of the reconstructed vehicle trajectory (green) for quantitative evaluation.

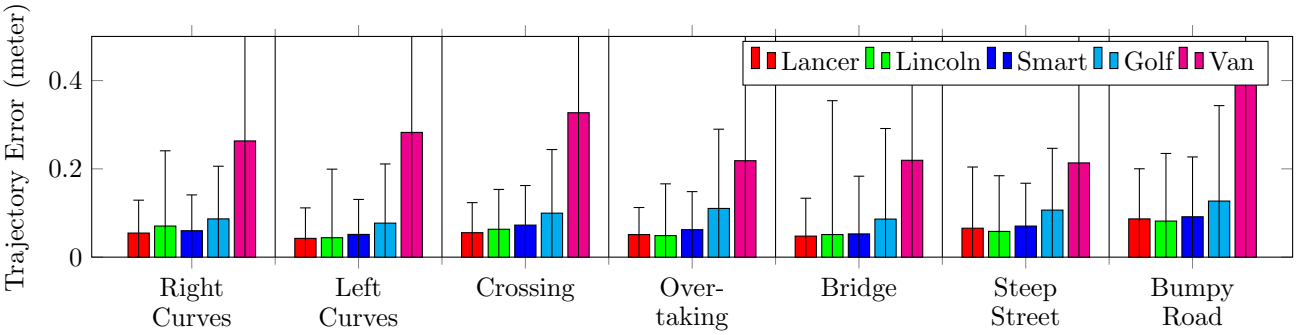


Figure 6: Quantitative evaluation of the proposed method using the dataset presented in [22]. The dataset contains seven different vehicle trajectories (*Right Curves*, *Left Curves*, *Crossing* ...) and five different vehicle models (*Lancer*, *Lincoln Navigator*, ...). The figure shows the trajectory error in meter, which is the average trajectory-point-mesh distance, i.e. the shortest distance of each object point to the vehicle mesh at the corresponding time step. The trajectory error is affected by background camera poses registration errors and incorrect vehicle point triangulations. The intervals shows the standard deviations of the trajectory error.

	Average Trajectory Error (meter)				
Method	Lancer	Lincoln	Smart	Van	Golf
Ours	**0.06**	**0.06**	**0.07**	**0.10**	**0.27**
[26]	0.11	0.09	0.14	0.21	0.30
[22]	0.20	0.23	0.33	0.33	0.47

Table 1: Trajectory error per vehicle of the benchmark dataset presented in [22]. Our approach achieves an average trajectory error of 0.11 m considering all sequences and outperforms the method presented in [22, 26].

of 5 different vehicles using a stereo camera baseline of 0.3 meter. The provided ground truth vehicle meshes allow to assess the trajectory at each frame of the sequence. We register the reconstructed object trajectory with the virtual environment by estimating a similarity transformation [25] between the reconstructed and the ground truth camera poses. Fig. 5 shows an example of a registered trajectory. For evaluation we use the trajectory error defined in [22], which is the shortest distance of each object point to the ground truth vehicle mesh of the corresponding time step. We achieve an average error of 0.11 meter and outperform the monocular methods presented in [22, 26]. Fig. 6 and table 1 show the corresponding results. Note that, [22, 26] apply SfM to the left images of the stereo camera to compute an object reconstruction, which shows a lower point density than the object point cloud obtained by stereo matching. We observe that the trajectory error of one vehicle is systematically worse than the average. A reason for this is that the Van vehicle model shows predominantly homogeneous surfaces, which hampers the stereo matching reconstruction quality. Our processing chain is not runtime-optimized and does currently not allow to process sequences in real time.

4 Conclusion

We presented a method to reconstruct three-dimensional object trajectories in stereo video data. Leveraging instance-aware semantic segmentation and optical flow techniques for Multiple Object Tracking allows us to determine object specific disparity values on pixel level. This avoids the triangulation of background structures while reconstructing object shapes using stereo matching. We evaluated our algorithm qualitatively on the KITTI and the Cityscapes dataset and outperformed previously published monocular trajectory reconstruction approaches on synthetic data of vehicles in an urban environment. In future work, we

intend to tackle the problem of invalid stereo matches caused by strong surface reflections.

References

[1] D. Scharstein and R. Szeliski, "A taxonomy and evaluation of dense two-frame stereo correspondence algorithms," *International Journal of Computer Vision*, 2002.

[2] S. Baker, S. Roth, D. Scharstein, M. J. Black, J. P. Lewis, and R. Szeliski, "A database and evaluation methodology for optical flow," in *2007 IEEE 11th International Conference on Computer Vision*, pp. 1–8, 2007.

[3] H. Hirschmuller, "Stereo processing by semiglobal matching and mutual information," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 30, pp. 328–341, Feb. 2008.

[4] A. Geiger, M. Roser, and R. Urtasun, "Efficient large-scale stereo matching," in *Asian Conference on Computer Vision (ACCV)*, 2010.

[5] K. Yamaguchi, D. McAllester, and R. Urtasun, "Efficient joint segmentation, occlusion labeling, stereo and flow estimation," in *Computer Vision – ECCV 2014* (D. Fleet, T. Pajdla, B. Schiele, and T. Tuytelaars, eds.), (Cham), pp. 756–771, Springer International Publishing, 2014.

[6] Z. Liang, Y. Feng, Y. Guo, H. Liu, W. Chen, L. Qiao, L. Zhou, and J. Zhang, "Learning for disparity estimation through feature constancy," in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018.

[7] J.-R. Chang and Y.-S. Chen, "Pyramid stereo matching network," in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018.

[8] "Robust vision challenge," 2018. [Online; accessed December 2018].

[9] L. Ladický, P. Sturgess, C. Russell, S. Sengupta, Y. Bastanlar, W. Clocksin, and P. H. S. Torr, "Joint optimization for object class segmentation and dense stereo reconstruction," *International Journal of Computer Vision*, vol. 100, pp. 122–133, Nov 2012.

[10] M. Bleyer, C. Rother, P. Kohli, D. Scharstein, and S. Sinha, "Object stereo - joint stereo matching and object segmentation," in *CVPR 2011*, pp. 3081–3088, June 2011.

[11] A. Ošep, W. Mehner, M. Mathias, and B. Leibe, "Combined image- and world-space tracking in traffic scenes," in *ICRA*, 2017.

[12] M. Coenen, F. Rottensteiner, and C. Heipke, "Recovering the 3d pose and shape of vehicles from stereo images," *Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, vol. IV, June 2018.

[13] F. Engelmann, J. Stückler, and B. Leibe, "SAMP: shape and motion priors for 4d vehicle reconstruction," in *IEEE Winter Conference on Applications of Computer Vision, WACV*, 2017.

[14] X. Chen, K. Kundu, Y. Zhu, A. G. Berneshawi, H. Ma, S. Fidler, and R. Urtasun, "3d object proposals for accurate object class detection," in *Advances in Neural Information Processing Systems 28* (C. Cortes, N. D. Lawrence, D. D. Lee, M. Sugiyama, and R. Garnett, eds.), pp. 424–432, Curran Associates, Inc., 2015.

[15] 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.

[16] A. Geiger, P. Lenz, C. Stiller, and R. Urtasun, "Vision meets robotics: The kitti dataset," *Int. J. Rob. Res.*, vol. 32, no. 11, 2013.

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

[18] P. Moulon, P. Monasse, R. Marlet, and Others, "Openmvg. an open multiple view geometry library.," 2013.

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

[20] 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)*, 2016.

[21] C. Chang and S. Chatterjee, "Quantization error analysis in stereo vision," in *[1992] Conference Record of the Twenty-Sixth Asilomar Conference on Signals, Systems Computers*, pp. 1037–1041 vol.2, Oct 1992.

[22] 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.

[23] C. Wu, "Visualsfm: A visual structure from motion system," 2011.

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

[25] S. Umeyama, "Least-squares estimation of transformation parameters between two point patterns.," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 13, no. 4, pp. 376–380, 1991.

[26] S. Bullinger, C. Bodensteiner, and M. Arens, "Monocular 3d vehicle trajectory reconstruction using terrain shape constraints," in *IEEE International Conference on Intelligent Transportation Systems (ITSC)*, 2018.