DeepKalPose: An enhanced deep-learning Kalman filter for temporally consistent monocular vehicle pose estimation

Leandro Di Bella,[1,2] Yangxintong Lyu,[1,2,✉] and Adrian Munteanu[1,2]

[1] *Department of Electronics and Informatics, Vrije Universiteit Brussel, Brussels, Belgium*

[2] *IMEC, Leuven, Belgium*

✉ E-mail: yangxintong.lyu@vub.be

This paper presents DeepKalPose, a novel approach for enhancing temporal consistency in monocular vehicle pose estimation applied on video through a deep-learning-based Kalman filter. By integrating a bi-directional Kalman filter strategy utilizing forward and backward time-series processing, combined with a learnable motion model to represent complex motion patterns, the method significantly improves pose accuracy and robustness across various conditions, particularly for occluded or distant vehicles. Experimental validation on the KITTI dataset confirms that DeepKalPose outperforms existing methods in both pose accuracy and temporal consistency.

Introduction: In recent years, the importance of scene understanding has become increasingly important, particularly in the development of technologies for smart mobility and intelligent transportation systems. The need to comprehend dynamic urban environments underscores the growing need for accurate video-based vehicle 6D pose estimation as real-world environments inherently give rise to dynamic data, such as video streams. However, traditional approaches [1–4] in this domain have primarily focused on image-based methodologies, which often yield temporal pose inconsistencies when applied to video data, processing each frame independently. These inconsistencies manifest as flickering artefacts, where vehicles exhibit jittery or unstable poses across successive frames. This phenomenon is primarily attributed to the absence of temporal constraints and coherency in the estimation process, significantly affecting scenarios involving occluded or distant vehicles leading to huge differences in predicted pose. This means they do not account for the continuity between frames in a video. To address this challenge, a range of innovative methods have been developed in the field of vehicle pose estimation that typically incorporate mechanisms to ensure temporal consistency, thereby reducing such artefacts.

A large body of research [5–7] has focused on shape completion where candidate shapes are encoded and compared against a predicted model shape, imposing temporal constraint. Yet, these methods often depend on additional data such as CAD models or dense point clouds, which are not always available. More recent works [8–10] propose to impose temporal relationship between frames by leveraging an LSTM-based model for motion learning. Nevertheless, LSTM approaches can suffer from challenges related to far-away detection as well as long-term dependencies. In contrast, particle filter-based methods, like Poisson multi-Bernoulli mixture (PMBM) tracking filter [11], have demonstrated robustness against such challenges. Moreover, [12–15] integrate a 3D Kalman filter (KF) into their frameworks, leveraging the kinematic motion of vehicles for smoothing and tracking tasks. However, challenges arise when dealing with systems where the underlying dynamics are unknown or have highly nonlinear behaviour. Under these conditions, creating a mathematical model for motion becomes very complex.

In this letter, we propose an enhanced deep-learning Kalman filter-based method (EDLFK) for temporally consistent vehicle pose estimation, dubbed DeepKalPose which is adept at reinforcing temporal consistency that overcomes the mentioned drawbacks. It can effectively address flickering artefacts in vehicle pose estimation by providing more stable and consistent tracking of the vehicle's pose. Our method advances beyond the standard KF by incorporating a bi-branch network for forward and backward time-series analysis. Additionally, our filter integrates deep learning techniques to effectively handle the nonlinear and complex motion patterns of vehicles. By employing an encoder–decoder architecture motion model, our method aims to design a more nuanced representation of vehicular dynamics. To summarize, our contributions are the following:

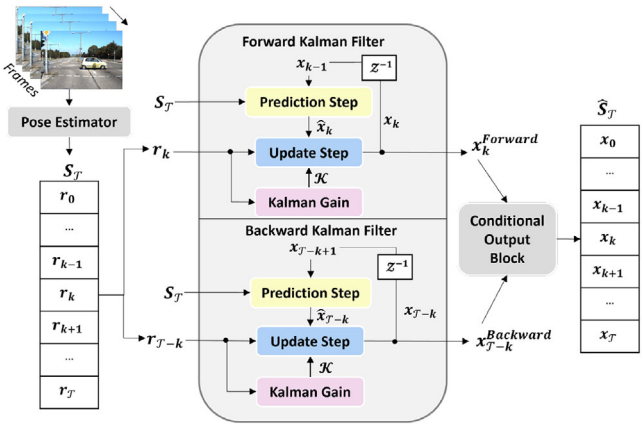

Fig. 1 *Schematic overview of DeepKalPose*

- We introduce a novel bi-directional KF strategy for offline vehicle pose smoothing, which employs both forward and backward processing, allowing an image-based pose estimator to process video.
- We propose a learnable motion model integrated into our Kalman filter, thereby enabling the network to learn more complex and nonlinear vehicle motion.
- Comprehensive experimental validation demonstrates that the proposed method outperforms the existing image-based techniques by adding temporal consistency, leading to robustness against occlusion and distant vehicles.

Notations: DeepKalPose estimates and adjusts vehicle pose estimates over sequential frames given the predictions of an existing pose estimator. Given the importance of the yaw angle in vehicle pose estimation, this letter focuses exclusively on the yaw angle within the rotation components. Our dataset, denoted as $\mathcal{D}$, comprises a series of samples $S_{i,\mathcal{T}}$, each representing a time-series of vehicle poses. $S_{i,\mathcal{T}}$ can be formally expressed as:

$$S_{i,\mathcal{T}} = \{\boldsymbol{r}_{i,k}\}_{k=1}^{\mathcal{T}} \tag{1}$$

Here, $\mathcal{D} = \{S_{i,\mathcal{T}}\}_{i=1}^{N}$ where $i \in \{1, \ldots, N\}$ with N representing the total number of samples. The variable $k \in \{1, \ldots, \mathcal{T}\}$ denotes the discrete time steps with $\mathcal{T}$ being a fixed time length of the time-series $S_{i,\mathcal{T}}$. A vehicle pose is represented as $\boldsymbol{r}_{i,k} = [x_{i,k}, y_{i,k}, z_{i,k}, \theta_{i,k}] \in \mathbb{R}^4$ where $x_{i,k}, y_{i,k}, z_{i,k}$ are the 3D translation components of the ith vehicle sample at timestep k and $\theta_{i,k}$ denotes the yaw or heading angle component.

Proposed method: An overview of the methodology is illustrated in Figure 1. This method will refine the pose predictions of an existing pose estimator to produce a more accurate and temporally consistent pose time-series output $\hat{S}_{\mathcal{T}}$ using KF algorithm; Refer to [16] for a comprehensive explanation of the Kalman filter algorithm. To achieve this, the filter takes the inconsistent pose prediction sequence $S_{\mathcal{T}}$ as input. At each iteration k, the measurement vector is updated as $\boldsymbol{r}_k = [\bar{x}_k, \bar{y}_k, \bar{z}_k, \bar{\theta}_k] \in S_{\mathcal{T}}$. For clarity, the symbol $\overline{(\cdot)}$ is used to denote measurements belonging to $S_{\mathcal{T}}$, differentiating them from the state vector used in the KF. Given the objective of tracking and correcting pose measurements, the state vector is defined as $\boldsymbol{x}_k = [x_k, y_k, z_k, \theta_k]$, thereby directing the KF to track the vehicle's pose. However, existing model-based KF methods [12, 13, 17, 18] have difficulties solving the temporal consistency since most time-series data from the autonomous driving dataset typically depict vehicles approaching or receding from the camera. Thus, the precision of pose estimation is significantly influenced by the vehicle's position within the image. Specifically, the accuracy of the estimates is higher for vehicles positioned closer to the camera than for vehicles farther away. This variability in accuracy impacts the effectiveness of KF tracking as the initial

Algorithm 1 Overview of DeepKalPose

Input: Complete trajectory data $\mathbf{T}$ of a vehicle

Output: Corrected trajectory data $\hat{\mathbf{T}}_{\mathbf{sub}} = \{\hat{S}_{i,\mathcal{T}}\}_{i=1}^{N}$

Initialization: Segment $\mathbf{T}$ into N sub-sequences $S_{i,\mathcal{T}}$ with fixed length $\mathcal{T}$ using stride one: $\mathbf{T}_{\mathbf{sub}} = \{S_{i,\mathcal{T}}\}_{i=1}^{N}$, where $i \in \{1, \ldots, N\}$

for $i = 1$ to N **do**

Data Pre-processing:

Initialize forward sequence $S^f_{i,\mathcal{T}} = [\mathbf{r}_{i,1}, \ldots, \mathbf{r}_{i,\mathcal{T}}]$, where $\mathbf{r}_{i,j}$ is the mean value of the valid poses if it is mis-detected

Initialize backward sequence similar to step 3, $S^b_{i,\mathcal{T}} = [\mathbf{r}_{i,\mathcal{T}}, \ldots, \mathbf{r}_{i,1}]$

Set initial states $\boldsymbol{x}^f = \boldsymbol{x}^b = \mathbf{0}$

Inference:

for $k = 0$ to $\mathcal{T}$ **do**

$\boldsymbol{x}^f[k] = \text{EDLKF}^f(S^f_{i,\mathcal{T}}[k], S^f_{i,\mathcal{T}})$

$\boldsymbol{x}^b[k] = \text{EDLKF}^b(S^b_{i,\mathcal{T}}[k], S^b_{i,\mathcal{T}})$

end for

$\hat{S}_{i,\mathcal{T}} = \text{COB}[\boldsymbol{x}^f, \boldsymbol{x}^b]$

end for

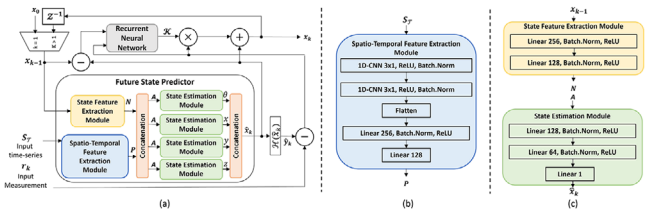

Fig. 2 *(a) Schematic of the proposed EDLKF architecture. (b) The spatio-temporal feature extraction module. (c) Details of the state feature extraction module (top) and state estimation module (bottom). $\mathcal{Z}^{-1}$ is a unit delay, $\otimes$ is multiplication operation, $\oplus$ is a sum operation and $\ominus$ is a substraction operation*

measurement and the corresponding precision play a crucial role in setting the starting point of the KF.

To address this challenge, we propose a bi-branch Kalman filter approach. The first branch, the forward KF, processes the time-series from timestep $k = 0$ to $\mathcal{T}$ where $\mathcal{T}$ is the context length of the input time sequence. Conversely, the backward KF branch processes the time-series in reverse, from the timestep $k = \mathcal{T}$ to 0. By selecting, through the conditional output block (COB), the most favourable samples from the time-series—closer to the camera at the initial step—we enhance the model's ability to accurately predict and track vehicle movements. The algorithm is outlined in Algorithm 1. We note that the proposed method acts as an offline smoother and processes the input data on chunks of $\mathcal{T}$ frames. While this approach does not render the method online, selecting sufficiently small $\mathcal{T}$ can enable near real-time processing. This method capitalizes on the higher quality of pose estimations near the camera, regardless of their position within the time-series. By feeding each time-series through both the forward and the backward EDLKF, our methodology aims to enhance the robustness and accuracy of pose estimations for distant and partially observed vehicles, as both scenarios present challenges for baseline estimations.

The Kalman filter necessitates a thorough understanding of the system's dynamics and noise characteristics to function effectively. However, for the pose estimation task, our limited knowledge of the vehicle's systems presents a significant challenge. Therefore, instead of using a traditional model-based Kalman filter, inspired by KalmanNet [19], our DeepKalPose replaces the computation the Kalman gain $\mathcal{K}$ and both of the covariances $\boldsymbol{v}_k$ and $\boldsymbol{w}_k$ by a recurrent neural network (RNN). Moreover, we propose a novel module, named future state predictor (FSP), which is able to learn a predictive motion model, as illustrated in Figure 2. The FSP block follows an encoder-decoder architecture where the encoder takes as input the state $\boldsymbol{x}_{k-1}$ and the sequence

*Table 1. Method comparison in terms of average precision. The methods denoted with * were retrained*

Method	Average precision @70			Average precision @50		
	Easy	Moderate	Hard	Easy	Moderate	Hard
D4LCN* [4]	28.07	21.56	14.13	67.08	52.55	35.74
Our method	**31.12**	**24.82**	**16.70**	**68.67**	**53.74**	**36.89**

that is being processed $S_{\mathcal{T}}$ with $\mathcal{T}$ the context length of the sequence and $k \in \{0, \ldots, \mathcal{T}\}$ representing the timestep of the KF iterative process. More precisely, the top branch of the encoder, namely the state feature extraction module (SFEM), processes the current state $\boldsymbol{x}_{k-1}$ to derive a feature vector $\mathbf{N}$. The SFEM captures the relevant aspects of the current state that may influence the future state. The lower branch of the encoder, namely spatio-temporal feature extraction module (STFEM) is dedicated to extracting features $\mathbf{P}$ from the entire sequence $S_{\mathcal{T}}$. This module analyzes local spatial and temporal patterns in the sequence of 3D position vectors, aiding in predicting the next state. Spatial patterns show the relationships among features like translation and rotation at each timestep, while temporal patterns track changes over time. In contrast, the SFEM focuses on immediate characteristics of the current state. The feature vectors from the SFEM and the STFEM are concatenated into a combined feature vector $\mathbf{A}$. The decoder consists of four state estimation modules (SEM) which will transform individually the feature vector $\mathbf{A}$ into the three translation components and the rotation component of the predicted state vector $\hat{\boldsymbol{x}}_k$. To train this model, we have used an L1-loss for the translation components as follows: $\mathcal{L}_\alpha = |\hat{\alpha} - \alpha|$ where $\alpha \in \{x, y, z\}$ and $\hat{\alpha}$ denotes the predicted translation values. The rotation loss is defined as: $\mathcal{L}_\theta = 1 - \cos(\hat{\theta}, \theta)$ where cos is the cosine similarity between the predicted heading angle $\hat{\theta}$ and the ground truth θ. The loss per mini-batch $\mathcal{B}$ with the mini-batch size $M < N$ is denoted as: $\mathcal{L}_\mathcal{B} = \frac{1}{M}\sum_{j=1}^{M}\frac{1}{\mathcal{T}}\sum_{k=0}^{\mathcal{T}}(\mathcal{L}_x + \mathcal{L}_y + \mathcal{L}_z + \mathcal{L}_\theta)_{j,k}$.

Experimental setup: The KITTI RAW dataset [20] is a widely used dataset for monocular object pose estimation [4, 21, 22] and tracking [23–25]. The dataset comprises 51 videos, divided into 39 for training and 12 for validation. It includes a total of 9903 images for training and 2977 images for validation. We segment each sequence into fixed length with 20 timesteps using a stride of 1 for continuity. In total, we have 674 vehicle trajectory patches for training and 375 for validation. To train DeepKalPose, we extract the pose predictions from an existing vehicle pose estimator [1, 4] as $\boldsymbol{r}_k$. To handle missing data from a non-detected vehicle by the vehicle pose estimator, we applied mean substitution. The network is optimized by Adam optimizer [26] with a learning rate of 0.001 and a weight decay of 0.00001. We take a batch size of 128 on 1 Nvidia GeForce RTX 2080 (12G). The iteration number for the training process is set to 4,000. Evaluation metrics used to compare against D4LCN [4] include 3D precision-recall curves with a 3D bounding box IoU threshold of 0.7 and 0.5 for cars as this method performs an object detection step. Against Mono6D [1], evaluation is performed using, for translation, the average relative Euclidean distance (**ARED**). For rotation, we use the accuracy with threshold δ, denoted **Acc**(δ), and the median error, **Mederr**, in degrees. Following the evaluation of D4LCN in [4], we only consider the detected vehicle with a 2D bounding box IoU threshold of 0.5.

Experimental results: In Table 1, we compared our proposed method, DeepKalPose, with the state-of-the-art (SOTA) method D4LCN [4]. One can note that with DeepKalPose, we can significantly outperform D4LCN [4]. The average precision @70 of D4LCN improves to 31.12% for Easy, 24.82% for moderate, and 16.70% for hard scenarios, compared to 28.07%, 21.56%, and 14.13% respectively without DeepKalPose. The same behaviour is observed for the AP@50.

In Table 2, the results illustrate the performance differences among the original Mono6D, Mono6D combined with a model-based Kalman filter (MB-KF), and Mono6D enhanced with DeepKalPose. The design of the MB-KF is inspired by the state-of-the-art tracking methodology presented in [25]. In the table, the best results are in bold and the second-best are underlined. One can observe that the integration of

*Table 2. Comparison with the existing methods. The methods denoted with * were retrained*

Method	ARED ↓	Acc $(\frac{\pi}{6})$ ↑	Acc $(\frac{\pi}{18})$ ↑	Mederr ↓
Mono6D* [1]	5.34%	84.66%	65.41%	**4.94°**
Mono6D* + MB-KF [25]	4.92%	84.69%	65.34%	5.03°
Mono6D + DeepKalPose (ours)	**3.90%**	**89.18%**	**67.47%**	5.46°

Table 3. Evaluation results at different occlusion levels

Method	Occlusion level	ARED ↓	Acc $(\frac{\pi}{6})$ ↑	Acc $(\frac{\pi}{18})$ ↑	Mederr ↓
Mono6D [1]	Visible	4.06%	92.35%	77.40%	**3.19°**
	Fully occluded	6.97%	73.86%	49.07%	10.19°
Our method	Visible	**3.64%**	**93.84%**	**79.00%**	4.21°
	Fully occluded	**4.21%**	**82.50%**	**51.53%**	**9.30°**

Table 4. Evaluation results at different depths

Method	Depth	ARED ↓	Acc $(\frac{\pi}{6})$ ↑	Acc $(\frac{\pi}{18})$ ↑	Mederr ↓
Mono6D [1]	0–40 m	5.18%	84.96%	65.86%	**4.85°**
	40m–∞	6.28%	82.94%	**62.81%**	**5.51°**
Our method	0–40 m	**3.96%**	**89.53%**	**68.29%**	5.24°
	40 m–∞	**3.55%**	**87.12%**	62.67%	7.02°

Table 5. Ablation studies on the KITTI validation set

STFEM	$\mathbf{B}_{\text{forw}}$	$\mathbf{B}_{\text{back}}$	ARED ↓	Acc $(\frac{\pi}{6})$ ↑	Mederr ↓
√	√		4.26%	88.78%	**5.10°**
	√	√	4.14%	84.79%	5.68°
√	√	√	**3.90%**	**89.18%**	5.46°

DeepKalPose led to notable improvements in vehicle 6D pose estimation. As such, the **ARED** sees an improvement from 5.34% to 3.90% while the MB-KF barely decreases the error. In addition, our method improves the heading angle orientation accuracy (**Acc**(δ)) by 4.52% for $\delta = \frac{\pi}{6}$ and 2.08% for $\delta = \frac{\pi}{18}$ although **Mederr** slightly increases, where the traditional method tends to replicate the results from the baseline. This result suggests that the EDL Kalman filter's smoothing operation leads to more stable rotations but at the expense of smaller, more frequent errors.

Furthermore, in scenarios of partial or complete vehicle occlusion in Table 3, this experiment reveals that DeepKalPose is robust against occlusion demonstrating a reduction of ARED error from 6.97% to 4.21%. Thanks to the predictive capabilities of the KF and the use of the temporal information from past measurements, DeepKalPose can adjust the initial noisy measurements from the pose estimator due to occlusion and put more emphasis on the learned predicted state vector. Moreover, in Table 4, we evaluate both methods for various vehicle distances from the camera. Specifically, for distances exceeding 40 m (considered as far-away-object detection), our proposed method demonstrates a notable decrease in ARED from 6.28% to 3.55%, indicating a 2.73% improvement. One can note that DeepKalPose utilizes past detections and temporal information effectively to mitigate the impact of distance on model performance, in contrast to the standard pose estimator whose performance diminishes with increasing distance. Figure 3 confirms the efficacy of DeepKalPose against far-vehicles pose estimation as illustrated by a bigger gap in ARED between the two methods when the distance is increasing. In Figure 4, we introduce a comparative visualization on a far-away and occluded vehicle highlighting the flickering artefacts present in the Mono6D estimates, where the 3D bounding boxes show significant rotational fluctuations over time. In contrast, our method introduces a temporal component that significantly enhances stability, with poses temporally coherent and reduced visual artefacts.

In Table 5, we mainly study the impact of the STFEM, and the impact of both the forward branch ($\mathbf{B}_{\text{forw}}$) and backward branch ($\mathbf{B}_{\text{back}}$).

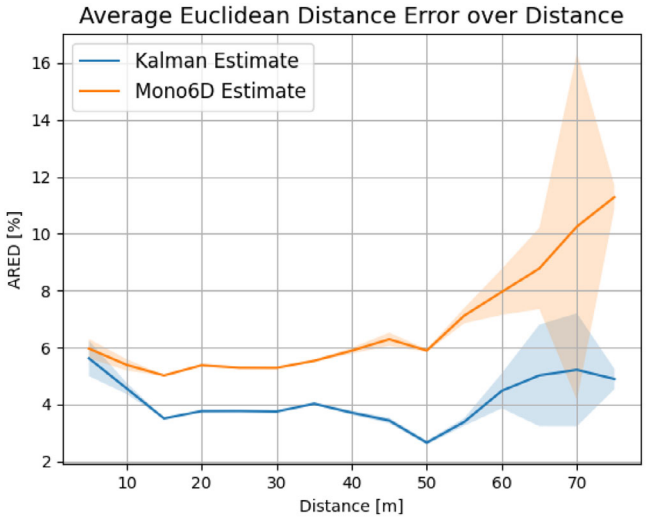

Fig. 3 *The ARED of Mono6D and proposed method against distance. Solid lines represent the mean values, while the shaded areas indicate the variance*

Fig. 4 *Qualitative results demonstrating improved car trajectory estimation on an occluded and distant vehicle. The sequence displays the projected 3D bounding boxes: the upper row illustrates results from Mono6D [1], while the lower row illustrates results from our proposed methodology. The two figures on the extreme right detail the temporal bird's eye view (BEV) comparison between the 3D BB estimations by Mono6D and DeepKalPose, respectively*

One can observe that for better translation performance, both directions should be taken into account. On top of that, Table 5 highlights the importance of the STFEM in the Kalman filtering process. A notable limitation of DeepKalPose is its operation as an offline algorithm, processing and analyzing the video data after it is fully processed by the pose estimator, which constrains its online applicability. While the method provides high accuracy and consistency in vehicle pose estimation, it's not ideal for scenarios where a frame needs to be processed immediately once captured. Future work will address this problem, moving from an offline method to an online method. Furthermore, for further improvements, we plan to replace the current empirical conditional output block (COB) with a deep-learning-based confidence network.

Conclusion: In this letter, we present DeepKalPose, an innovative approach integrating a deep learning-based Kalman filter to enhance temporal consistency in monocular vehicle 6D pose estimation for video data. Leveraging a learnable motion model, DeepKalPose effectively captures the complex, nonlinear motion patterns of vehicles, significantly surpassing existing methods in accuracy and consistency for 4D object detection and 6D pose estimation tasks. The experimental results demonstrate the model's effectiveness in improving pose estimation accuracy and consistency from single-view images, particularly in challenging conditions such as far-object detection and occlusion. These results affirm the efficacy of our deep-learning-based Kalman filter in video-based pose estimation and suggest its potential to enhance intelligent transport systems.

Author contributions: **Leandro Di Bella**: Conceptualization; data curation; formal analysis; investigation; methodology; software; validation; visualization; writing—original draft; writing—review and editing. **Yangxintong Lyu**: Conceptualization; methodology; project administration; supervision; validation; writing—review and editing.

Adrian Munteanu: Conceptualization; funding acquisition; project administration; resources; supervision; validation; writing—review and editing.

Acknowledgments: This work is funded by Innoviris within the research project TORRES.

Conflict of interest statement: The authors declare no conflicts of interest.

Data availability statement: The data that support the findings of this study are available in KITTI RAW Dataset at [https://www.cvlibs.net/datasets/kitti/], reference number [20].

Received: *23 January 2024* Accepted: *27 March 2024*
doi: 10.1049/ell2.13191

References

1 Lyu, Y., Royen, R., Munteanu, A.: MONO6D: monocular vehicle 6d pose estimation with 3D priors. In: 2022 IEEE International Conference on Image Processing (ICIP), pp. 2187–2191. IEEE, Piscataway, NJ (2022)
2 Ke, L., Li, S., Sun, Y., Tai, Y.-W., Tang, C.-K.: GSNet: joint vehicle pose and shape reconstruction with geometrical and scene-aware supervision. In: Computer Vision–ECCV 2020: 16th European Conference, pp. 515–532. Springer, Cham (2020)
3 Wu, D., Zhuang, Z., Xiang, C., Zou, W., Li, X.: 6D-VNet: end-to-end 6-DOF vehicle pose estimation from monocular RGB images. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops, pp. 1238–1247. IEEE, Piscataway, NJ (2019)
4 Ding, M., Huo, Y., Yi, H., Wang, Z., Shi, J., Lu, Z., Luo, P.: Learning depth-guided convolutions for monocular 3D object detection. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops, pp. 1000–1001. IEEE, Piscataway, NJ (2020)
5 Hödlmoser, M., Micusik, B., Pollefeys, M., Liu, M.-Y., Kampel, M.: Model-based vehicle pose estimation and tracking in videos using random forests. In: 2013 International Conference on 3D Vision-3DV 2013, pp. 430–437. IEEE, Piscataway, NJ (2013)
6 Giancola, S., Zarzar, J., Ghanem, B.: Leveraging shape completion for 3D Siamese tracking. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 1359–1368. IEEE, Piscataway, NJ (2019)
7 Sharma, S., Ansari, J.A., Murthy, J.K., Krishna, K.M.: Beyond pixels: Leveraging geometry and shape cues for online multi-object tracking. In: 2018 IEEE International Conference on Robotics and Automation (ICRA), pp. 3508–3515. IEEE, Piscataway, NJ (2018)
8 Hu, H.N., Yang, Y.-H., Fischer, T., Darrell, T., Yu, F., Sun, M.: Monocular quasi-dense 3D object tracking. *IEEE Trans. Pattern Anal. Mach. Intell.* **45**(2), 1992–2008 (2022)
9 Hu, H.N., Cai, Q.-Z., Wang, D., Lin, J., Sun, M., Krahenbuhl, P., Darrell, T., Yu, F.: Joint monocular 3D vehicle detection and tracking. In: Proceedings of the IEEE/CVF International Conference on Computer Vision, pp. 5390–5399. IEEE, Piscataway, NJ (2019)
10 Marinello, N., Proesmans, M., Van Gool, L.: Triplettrack: 3D object tracking using triplet embeddings and LSTM. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 4500–4510. IEEE, Piscataway, NJ (2022)
11 Scheidegger, S., Benjaminsson, J., Rosenberg, E., Krishnan, A., Granström, K.: Mono-camera 3D multi-object tracking using deep learning detections and PMBM filtering. In: 2018 IEEE Intelligent Vehicles Symposium (IV), pp. 433–440. IEEE, Piscataway, NJ (2018)
12 Brazil, G., Pons-Moll, G., Liu, X., Schiele, B.: Kinematic 3D object detection in monocular video. In: Computer Vision–ECCV 2020: 16th European Conference, pp. 135–152. Springer, Cham (2020)
13 Reich, A., Wuensche, H.J.: Monocular 3D multi-object tracking with an EKF approach for long-term stable tracks. In: 2021 IEEE 24th International Conference on Information Fusion (FUSION), pp. 1–7. IEEE, Piscataway, NJ (2021)
14 Wu, Y., Sheng, H., Zhang, Y., Wang, S., Xiong, Z., Ke, W.: Hybrid motion model for multiple object tracking in mobile devices. *IEEE Internet Things J.* **10**(6), 4735–4748 (2023)
15 Wang, L., Zhang, X., Qin, W., Li, X., Gao, J., Yang, L., Li, Z., Li, J., Zhu, L., Wang, H., et al.: CAMO-MOT: combined appearance-motion optimization for 3D multi-object tracking with camera-LIDAR fusion. *IEEE Trans. Intell. Transp. Syst*. **24**(11), 11981–11996 (2023)
16 Welch, G., Bishop, G., et al.: An introduction to the kalman filter. 1995.
17 Pak, J.: Visual odometry particle filter for improving accuracy of visual object trackers. *Electron. Lett.* **56**(17), 884–887 (2020)
18 Wu, Y., Vandewalle, P., Slaets, P., Demeester, E.: An improved approach to 6D object pose tracking in fast motion scenarios. In: 2022 Sixth IEEE International Conference on Robotic Computing (IRC), pp. 229–237. IEEE, Piscataway, NJ (2022)
19 Revach, G., Shlezinger, N., Ni, X., Escoriza, A.L., Van Sloun, R.J., Eldar, Y.C.: KalmanNet: neural network aided Kalman filtering for partially known dynamics. *IEEE Trans. Signal Process.* **70**, 1532–1547 (2022)
20 Geiger, A., Lenz, P., Urtasun, R.: Are we ready for autonomous driving? the KITTI vision benchmark suite. In: 2012 IEEE conference on computer vision and pattern recognition, pp. 3354–3361. IEEE, Piscataway, NJ (2012)
21 Zhang, Y., Lu, J., Zhou, J.: Objects are different: flexible monocular 3D object detection. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 3289–3298. IEEE, Piscataway, NJ (2021)
22 Peng, L., Wu, X., Yang, Z., Liu, H., Cai, D.: DID-M3D: decoupling instance depth for monocular 3D object detection. In: European Conference on Computer Vision, pp. 71–88. Springer, Cham (2022)
23 Cao, J., Pang, J., Weng, X., Khirodkar, R., Kitani, K.: Observation-centric sort: rethinking sort for robust multi-object tracking. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 9686–9696. IEEE, Piscataway, NJ (2023)
24 Kim, A., Brasó, G., Ošep, A., Leal-Taixé, L.: Polarmot: How far can geometric relations take us in 3D multi-object tracking? In: European Conference on Computer Vision, pp. 41–58. Springer, Cham (2022)
25 Wu, H., Han, W., Wen, C., Li, X., Wang, C.: 3D multi-object tracking in point clouds based on prediction confidence-guided data association. *IEEE Trans. Intell. Transp. Syst.* **23**(6), 5668–5677 (2021)
26 Kingma, D.P., Ba, J.: Adam: a method for stochastic optimization. arXiv:1412.6980 (2014)