An Efficient Multi-Object Tracking Framework for Autonomous Driving

Kemiao Huang[1] and Qi Hao[1]

***Abstract*— Online and real time are two fundamental requirements for autonomous driving. As an integral module in the perception system, multi-object tracking (MOT) should provide reliable object trajectories in 3D space. This paper proposes an efficient framework for online 3D MOT from multiple sensors by using both motion and appearance cues with a state-wise matching criterion. This framework is employed with state-of-the-art models and the quantitative results on the challenging KITTI benchmark shows its competitive performance. Available code: https://github.com/Kemo-Huang/EMOT**

I. INTRODUCTION

Multiple object tracking (MOT) is a crucial task in autonomous driving where the aim is to sequentially identify multiple traffic participants and enable further reasonings such as motion forecasting and collision avoidance. Tracking-by-detection is the current leading strategy for MOT due to the advance of object detectors in computer vision. This strategy has moved research attention to the data association problem, including how to define scoring metrics to represent the level of similarity between the objects as well as how to match objects across frames according to those metrics. Although extensive research has been done in these years, it remains challenging for real world employment due to the unconstrained environments. Many state-of-the-art methods [1] [2] focus on increasing the tracking accuracy by deep neural networks for modelling object appearance or motion on large-scale datasets. However, these discriminative networks cannot be well interpreted for online decision making and self-tuning in autonomous systems. Thus they are not very adaptable for complex environments such as crowded urban scenes.

Different from conventional challenges, autonomous driving has to manage the correlation between multi-modality inputs. Most of the prior tracking methods only focus on 2D visual inputs in a stable environment. However, depth sensors such as radar and LiDAR are more useful for 3D localization so point cloud tracking plays an important role. Moreover, autonomous vehicles are expected to run under all possible circumstances such as in dark night or rainy weather. These requirements force the perception methods to enable intelligent sensor fusion. In addition, autonomous driving has strict requirement upon the computational cost of perception algorithms. Processing all the sensor data in each frame will slow down the overall response speed of the system.

To tackle the above issues, we propose a robust online MOT framework with a polished multi-modality data association pipeline. Ideally, the proposed framework is extendable to various motion/appearance models, multiple sensors and different linear matching algorithms. To obtain better fusion between multiple modalities, the deep features from images and point cloud can be fused by employing weighted sum or extra neural networks. In this work, online motion filtering, pretrained deep feature networks and Hungarian algorithm [3] are used. Furthermore, the proposed method well balances the tradeoff between real time and accuracy by defining a state-wise matching criterion between measurements and predictions for both motion and appearance costs. This pipeline is more flexible than end-to-end deep neural networks without increasing computations.

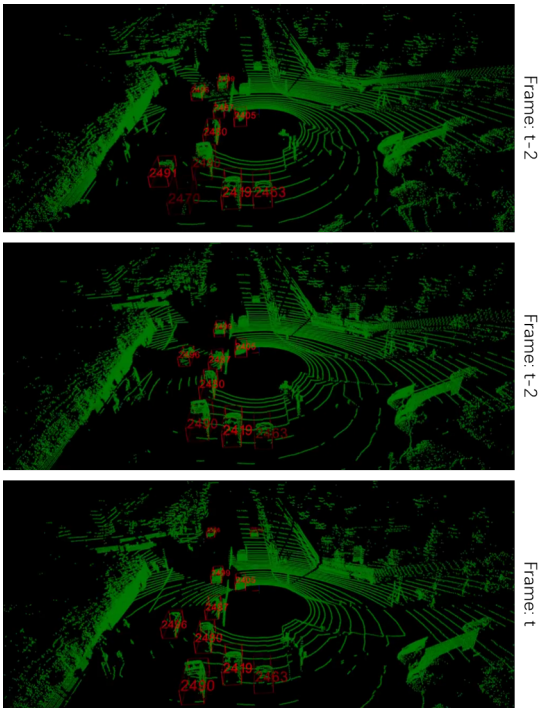

Fig. 1. 3D multi-vehicle tracking in real world driving environment. The crowded crossroads is one of the challenging scenarios including heavy occlusion and complex motions.

In summary, our contributions are as follows:

1. This work proposes an efficient and extendable MOT framework for online, real time and multi-modality applications.

2. This work combines the state-of-the-art appearance and motion modelling methods with online state-wise decisions for quick and robust data association.

3. This work employs and evaluates the proposed MOT framework on KITTI [4] MOT benchmark. The competitive

[1]The authors are with the Intelligent Sensing and Unmanned Systems Laboratory, Southern University of Science and Technology. 11610728@mail.sustech.edu.cn, hao.q@sustech.edu.cn

performance shows that our tracking framework is adaptable and reliable.

II. Related Work

A. Multi-Object Tracking Paradigm

Tracking-by-detection is the primary paradigm for recent MOT research due to the high accuracy of the detector. Some other state-of-art tracking paradigms are also studied. For example, [5] propose a tracking network for arbitrary spatial tracking which has no limitation to object class or existence. As autonomous driving is only interested in tracking objects of certain classes, we basically follow the tracking-by-detection paradigm in this work. Among the prior approaches, SORT [6] is one of the favorable baselines in academic communities and industries due to its high performance. We generalize it idea as a basic framework for tracking-by-detection, which is to apply motion filtering over the sequence to predict the tracking objects at current frame by posterior estimation and compute the overlaps of the predicted targets and the detection results to assign the identities by linear matching algorithms. The filtering methods are commonly Gaussian process [7], particle filter, Kalman filter [8] and Poisson multi-Bernoulli mixture (PMBM) filter [9]. Besides filters, batch methods such as recurrent neural networks (RNN) and long short-term memory (LSTM) are also used to encode the state space. In contrast to filtering methods, these approaches can demonstrate the target motion more flexibly but the parameters of such complex models are more challenging to be tuned [10]. As the advance of computer vision, appearance-based tracking is extensively experimented. Some state-of-the-art research [2] [11] use end-to-end model to directly produce identities from raw inputs. However, these methods only associate targets by the image features of each frame without considering sequential motion and bounding box overlaps, which is usually ineffective when the targets have similar appearance or severe occlusion.

B. Feature Extraction and Correlation

Although SORT-like tracking frameworks are fast and simple to be employed in most scenarios, the shortcomings are also obvious: its performance highly depends on the quality of the detector and the false negatives are propagated when the targets have sudden motion change after misdetection. To overcome this issue, [12] proposes a deep association metric to involve image information by using a convolutional neural network (CNN) which is trained on person re-identification dataset. The image feature extraction is relatively easy due to the extensive prior research, while extracting point cloud features remains challenging because of data sparsity and randomness. Traditional method is to measure distances [13] between points, convert to 2.5D grid [14] or other hand-crafted features [15]. To fully exploit the inherent information, PointNet [16] and PointNet++ [17] use symmetric functions to process raw points to provide deep representation, which are the primary methods applied in 3D object detection and semantic segmentation. After extracting features, the correlation functions for scoring similarities between detections and tracks should also be carefully designed. While most of prior methods [12] [2] focus on image level affinity estimation, [1] designs an attentional fusion module for exploiting both image and point cloud features and applies pointwise convolution on the features. In addition to appearance and motion information, [18] also uses the structure and size information from persons to compute the final distances across frames with and achieves high performance. However, the appearance and structure features are hand-crafted, which is not always suitable for various object classes.

C. Data association

The data association problem has been studied extensively for many years. Hungarian algorithm (HA) [3] is the most famous solution whose computational complexity is $O(N^3)$. In addition to HA, the data association problem can also be tackled by other methods such as Markov decision process (MDP) [19], min-cost flow [20] [2], joint probability data association (JPDA) [21], multiple hypothesis tracking (MHT) [22]. The recent end-to-end approaches [2] [1] try to train the models for optimal target assignment by deep neural networks and linear programming. In this work, we treat the target assignment task as a general NP-hard max-matching problem and just use HA to avoid hyperparameters and model overfitting.

D. Object Detection

The performance of object detectors is critical in tracking-by-detection paradigm. As a hit of deep approach in 2D object detection, Faster RCNN [23] uses VGG16 and region proposals to get high performance. On the other hand, as a 3D extension, [24] uses segmentation to generate point cloud proposals and refines them in two stages. [25] converts point cloud to bird's eye view and applies 2D convolutions to increase efficiency for predicting 3D detections. [26] aggregate image and point cloud features from different views. [27] gets frustum proposals from images and applies PointNet [16] on the point clouds within the frustums.

III. Multi-Object Tracking

This objective of this MOT is to sequentially identify the multiple detections of each frame by exploiting both motion and appearance information with a matching policy. Following the common tracking-by-detection paradigm, the in proposed MOT framework can be divided into five modules, namely object detection, motion filtering, feature extraction, data association and state management, as shown in Figure 2. First of all, the object detector localizes the object with bounding boxes and their confidence. Second, the 3D Kalman filter predicts the track motion states at the current frame. Then, the matching module assignment each detections with existing tracks by the overlap of bounding boxes for initial data association. According to the matching results and the tracks states, further association is done on the appearance feature level. The 2D and 3D features of

the tentative targets are extracted separately from image and point cloud within the bounding boxes. The unmatched detections and tracks are associated by their appearance cues to reduce false negatives. The final association results are used to update the track states as well as their matching criterions.

A. Problem Formulation

Since our tracking system is based on multiple modalities, the object detector should provide both 2D and 3D bounding boxes in both image and world coordinates. The transformation of bounding boxes in different sensors should be done in detection stage. In this work, all sensors are assumed to be calibrated and synchronized at each frame so that the coordinate transformation is viewed as noise-free. The input data of MOT at the i^{th} frame is a set of detections and the sensor data:

$$D^i = \{d^i_j \mid j = 1, \ldots, N\}, \tag{1}$$

where N is the number of detections. Specifically, the detections consist of the bounding boxes, confidence and the cropped image patches and point cloud.

$$d_{ij} = \begin{cases} x_1, y_1, x_2, y_2, & 2D\ bbox \\ x, y, z, l, w, h, \theta, & 3D\ bbox \\ s, & confidence \\ im, & image\ patch \\ pc, & point\ cloud \end{cases} \tag{2}$$

where (x_1, y_1, x_2, y_2) are the 2D coordinates, (x, y, z) are the 3D coordinates, (l, w, h) are the box size and θ is the yaw angle for the orientation. If the detector does not provide confidence score, we can generate it at the later feature extraction module. In our MOT framework, the data association cost is defined by the correlation metric but there is no limitation on the way to solve the assignment problem. In Hungarian algorithm [3], it is treated as a weighted bipartite graph matching problem or a global nearest neighbor (GNN) problem. The costs between two graph nodes are estimated in section III-D.

B. Motion Filtering

In this work, the motion of the object is simply approximated to a linear constant velocity model across frames. To realize motion modelling, Kalman filter (KF) is used to propagate the locations with statistical noise. The state variable is modelled as a combination of the detection data with the additional linear velocity in 3D space: $X = (x, y, z, l, w, h, \theta, s, v_x, v_y, v_z)$ Although the bounding box size, orientation and the detection confidence in measurement do not contribute to the object motion, we propose to reserve them in the state variable to smooth the detections. Note that the other transition parameters such as the angular velocity and the acceleration can also be addressed in KF but this study does not include them because more errors are observed. In state prediction, the transition of the location variables across frames is linear to time. Specifically, the estimate in the current frame can be derived by the location and velocity from the previous frame with equations:

$$\begin{cases} x_i = x_{i-1} + v_x \times dt \\ y_i = y_{i-1} + v_y \times dt \\ z_i = z_{i-1} + v_z \times dt \end{cases} \tag{3}$$

where $dt = 1$ between each frame. In state update, the updated target states are the weighted sum of the associated measurements and predicted targets according to their uncertainties. The mechanism of KF is depicted in [8]. In general, the motion model predicts the approximate positions, sizes and orientations of the bounding boxes in the next frame, which makes it possible to compute the spatial overlap of the detections and tracks.

C. Feature Extraction

To exploit the appearance information, the target features are extracted from the corresponding image and point cloud within the bounding boxes. Note that the appearance features for tracking targets can be adapted from object detectors in some joint detection/tracking learning models [28] but this work uses the conventional tracking-by-detection cascade so we need to construct new models to extract features from given detections. Many off-the-shelf models in computer vision include feature extractors of high performance. In this work, we define three strategies to estimate appearance distances from multiple modalities: early fusion, middle fusion and late fusion. Early fusion is to fuse the raw data from different sensors at a preprocessing stage and generate region proposals to extract features. Middle fusion is to fuse the aligned multi-modality features from extracted from separate sensors and define correlation between the fused features. Late fusion is to use the weighted sum of the independent feature distances as the final appearance distance. As there is no standard length for the LiDAR inputs, region proposal networks from typical image processing are not suitable for fusing images and point cloud. However, we agree that deep learning is highly effective for feature fusion and more optimal than postprocessing approaches for normal situations. In this work, we modify the model in [1] as a middle-fusion approach to extract robust multi-modality features. The feature dimensions defined in [1] are different from this work because we do not simultaneously extract features from two or more frames for linking. We only extract features from current frame and estimate the distance between the new features and the features in tracks' memory.

D. Data Association and State Management

1) Distance Estimation: The cost matrix for the data association algorithm is defined by the bounding box distance and the appearance feature distance. The bounding boxes include the basic spatial information of the targets and the intersection-over-union (IoU) can effectively address the location and size similarity between targets. The bounding box distance is defined as one minus the IoU.

$$E_{iou} = 1.0 - \frac{b_1 \cap b_2}{b_1 \cup b_2} \tag{4}$$

The effectiveness of this cost highly depends on the quality of the bounding boxes provided by the object detector. To increase the system robustness against complex motion, appearance features should be used. The correlation function for the extracted features is the point-wise "absolute subtraction".

$$corr(a, b) = abs(a - b) \tag{5}$$

The appearance-based distance is measured by taking the minimum value of the correlation between detected features and k history tracking features.

$$E_{feat} = \min_i \sum_n corr(feat(d), feat^i(t)), i = 1, \ldots, k \tag{6}$$

Where n is the feature length, $feat(d)$ is the feature from the d^{th} detection and $feat^i(t)$ is the i^{th} feature from the t^{th} track.

2) Matching Criterion: As appearance feature extraction is usually time-consuming, we propose a state-wise matching pipeline to reduce the computational complexity without losing necessary information. Totally four target states are defined in this work: $birth$, $steady$, $motion\ lost$, $occluded$ and $death$. Only the steady tracks broadcast their identities and only the dead tracks stop making predictions. Matched objects can grow into steady tracks and unmatched tracks are deleted only if their hit times and miss times reach the thresholds, respectively. These thresholds: growth time λ_1 and decay time λ_2 are defined according to the detection confidence or feature confidence:

$$\lambda_1 = floor(\alpha \times (1 - s)), s \in [0, 1] \tag{7}$$

$$\lambda_2 = floor(\beta \times s), s \in [0, 1] \tag{8}$$

Where α is the maximum growth time and β is the maximum decay time.

The matching criterion and pipeline are depicted in algorithm III-D.2. Firstly, the IoU costs between measured objects and predicted objects are used as the matching cost for Hungarian algorithm and the matched outputs are filtered with an IoU threshold θ_{iou}. We regard the unmatched measurements in the first stage as the motion-lost tracks. Then, the features of the unmatched measurements are extracted and the unmatched objects are further associated by using the feature costs. Same as the previous association, the matched pairs with feature distance larger than a threshold θ_{feat} are filtered. The unmatched objects in the second stage are regarded as occluded or dead. Using IoU association before appearance association can quickly find out many obvious matches and save time. Furthermore, in order to avoid feature disappearance for tracks, the features are extracted for all newcomers to keep the good "first impressions". To tackle the appearance change, the features are updated intermittently with a sampling time p and the past features are stored in tracks' history lists of length k.

Algorithm 1 Efficient Association (D, T)

if $T = \varnothing$ **then**
  $U'_d \leftarrow D$
  jump to line 21
end if
$P \leftarrow predict(T)$, $T.miss$++
$M, U_d, U_t \leftarrow Hungarian(E_{iou}, D, P)$ (eq.4)
for $d, t \in M$ **do**
  $Kalman(d, t)$ (eq.3)
  $t.skip$++, $t.hit$++, $t.miss \leftarrow 0$
  if $t.skip \geq p$ **then**
    push $feat(d)$ into queue $t.history$ of size k
    update t by λ_1, λ_2 (eq.7, 8)
    $t.skip \leftarrow 0$
  end if
end for
$M', U'_d, U'_t \leftarrow Hungarian(E_{feat}, U_d, P)$ (eq.6)
for $d, t \in M'$ **do**
  $Kalman(d, t)$
  update t by $feat(d), \lambda_1, \lambda_2$
end for
for $d \in U'_d$ **do**
  create
  ectory t by $feat(d), \lambda_1, \lambda_2$
  $T \cup t$
end for
for $t \in T$ **do**
  if $t.hit \geq t.\lambda_1$ and $t.miss \leq t.\lambda_2$ **then**
    $T_{out} \cup t$
  else if $t.miss > t.\lambda_2$ **then**
    delete t from T
  end if
end for
return T_{out}

IV. Experiments

A. Datasets

The tracking system is evaluated on the challenging KITTI Tracking Benchmark [4], which consists of 21 training sequences and 29 testing sequences. LiDAR point cloud, RGB images as well as calibration files are provided for each sequence. There are 30601 annotated objects and 636 tracks in the training sequences, including cars, pedestrians, cyclists, etc. In this work, we use 10 sequences from the training sequences as training set and the remaining 11 sequences as validation set. The total number of frames is 3975 for training and 3945 for validation. This training/validation set split is the same as others' work [1] for fair comparison. Besides, we only evaluate the "Car" set for simplicity.

B. Implementation Details

For the first detection stage, we select the open-source PointPillars [25] detector to get the 3D detection results. The ID labels of the ground truth bounding boxes are assigned to detections with the greatest IoU with threshold 0.5. The deep

TABLE I

3D MOT QUANTITATIVE RESULTS ON THE VALIDATION SET OF KITTI MOT BENCHMARK

Method	MOTA↑	MOTP↑	Prec.↑	Recall↑	FP↓	FN↓	ID-s↓	Frag↓	MT↑	ML↓	FPS↑
AB3DMOT	69.22	**86.76**	85.78	89.18	1975	1446	**0**	**71**	75.46	3.7	203.4
mmMOT	**79.46**	85.44	93.45	**88.84**	791	**1417**	75	272	**76.85**	**2.32**	13.34
ours(motion)	74.34	85.72	**94.73**	81.93	**572**	2270	10	75	59.26	18.52	**240.2**
ours(motion+appearance)	77.71	85.68	94.12	85.80	676	1791	10	96	63.43	5.56	42.71

learning modules are implemented by PyTorch [29] upon NVIDIA Tesla M40. Specifically, we follow the front part of the network architecture in [1] for single modality feature extraction and multi-modality fusion. The image feature extractor is based on VGGNet [30] with skip-pooling [31] and the point feature extractor is based on PointNet [16] with average pooling for each detection. The linear assignment algorithm is implemented by using scikit-learn [32].

C. Evaluation Metrics

We follow the standard MOT metrics in CLEAR MOT [33] including identity switches, overall accuracy, precision, mostly tracked trajectories, mostly lost trajectories and average running time. We follow the data format and the evaluation code provided by KITTI benchmark [4] for each method.

D. Experimental Results

The overall quantitative results are shown in Table I. For comparison, we experiment the open-source methods [34] and [1] on the same validation set. Our own implementation of the motion-based approach has both higher accuracy and speed than the baseline [34]. What's more, the proposed mixture model successfully increase the tracking speed without losing much accuracy.

TABLE II

COMPARISON OF DIFFERENT COST THRESHOLDS FOR DATA ASSOCIATION.

θ_{iou}	θ_{feat}	MOTA↑	MOTP↑
0.01	5	76.90	**85.71**
0.01	10	**77.14**	85.42
0.01	20	76.02	85.65
0	10	76.88	85.40
0.001	10	**77.40**	85.42
0.05	10	76.47	85.42
0.5	10	43.61	**85.57**

Since our framework is highly modular, we experiment with different configurations for ablation study. There are totally six hyperparameters in this work: $\{\theta_{iou}, \theta_{feat}, \alpha, \beta, p, k\}$. Firstly, we suggest different cost thresholds in IoU and feature association modules to test the efficiency of motion and appearance cues. As we choose IoU association for the first matching stage, the highest tracking accuracy is subject to the performance of motion model. From the results in Table II, both motion and appearance models matter in the proposed framework. Not surprisingly, optimal thresholds are relatively small values. However, small IoU threshold restricts the matching results from including targets with complex motions or velocity changes after sudden occlusion and small feature threshold discards the objects with partial occlusion.

TABLE III

COMPARISON OF DIFFERENT GROWTH TIME AND DECAY TIME.

α	β	MOTA↑	MOTP↑	FP↓	FN↓
4	2	76.36	85.42	**901**	1412
2	2	**76.38**	85.42	903	1410
2	3	72.43	85.07	1680	**1189**

The track birth time and decay time control the states of the targets. The results in Table III show that the growth time does not have significant influence on the performance but the decay time matters. This is because the slow "growth" procedure only ignores a part of false positives and sudden disappeared objects but the slow "decay" procedure keeps the potential tracks to tackle the occlusion problem. The drawback of slow "decay" is that it does not only increase system robustness against occlusion but also increase the false positives by linking other objects being close to the original ones or objects with similar looking.

TABLE IV

COMPARISON OF DIFFERENT HISTORY LENGTH FOR TRACKS' FEATURES.

k	MOTA↑	MOTP↑	FP↓	FN↓	FPS↑
2	77.14	85.42	908	1411	**34.31**
4	77.14	85.42	908	**1410**	35.35

The homogeneous results for different feature memory sizes k in Table IV confirm that the most time consuming module is the feature extraction and the feature correlation module does not increase much complexity. However, increasing different memory sizes does not significantly improve the performance. The ineffectiveness of history features is probably due to the fact that the objects in dataset seldomly return to their previous appearance states.

TABLE V

COMPARISON OF SAMPLING TIME FOR TRACK UPDATE.

p	MOTA↑	MOTP↑	FP↓	FN↓	FPS↑
1	**76.88**	85.40	936	**1392**	12.13
10	73.59	**85.42**	**892**	1417	**37.42**

According to Table V, larger sampling intervals significantly reduce the running time, which again reflects the high complexity of feature extraction. However, reducing feature update frequency has the drawback of losing online appearance information. The fast vehicles' relative motions

can produce large affinity transforms which increase the classification difficulties. We need to choose a suitable sampling time to tradeoff the accuracy and computational cost.

Hyperparameter tuning is a crucial task in the proposed framework. Bayesian optimization is a common and useful strategy for global optimization for modular functions without derivatives. We can use this strategy to further optimize the matching pipeline. The details about the optimization approaches are described in [35], which are beyond this project.

V. Conclusion

This work proposes an online 3D MOT framework for autonomous vehicles with visual and depth sensors. This framework focuses on both appearance and motion cues with a state-wise data matching pipeline and it is adaptable to most machine learning models. We implement the proposed tracker with state-of-the-art feature extractors and 3D Kalman filter. The hyperparameters are tuned to tradeoff saving computational costs and increasing matching pipeline robustness. The competitive results on KITTI MOT benchmark shows the high efficiency of both motion-based and mixture model-based approaches. By introducing more flexible motion models and optimal association algorithms can further improve the tracking accuracy.

References

[1] Z. Wenwei, Z. Hui, S. Shuyang, W. Zhe, S. Jianping, and L. C. Change, "Robust multi-modality multi-object tracking," in *The IEEE International Conference on Computer Vision (ICCV)*, October 2019.

[2] D. Frossard and R. Urtasun, "End-to-end learning of multi-sensor 3d tracking by detection," in *2018 IEEE International Conference on Robotics and Automation (ICRA)*, May 2018, pp. 635–642.

[3] P. Emami, P. Pardalos, L. Elefteriadou, and S. Ranka, "Machine learning methods for solving assignment problems in multi-target tracking," Feburary 2018.

[4] A. Geiger, P. Lenz, and R. Urtasun, "Are we ready for autonomous driving? the kitti vision benchmark suite," in *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2012.

[5] A. R. Kosiorek, A. Bewley, and I. Posner, "Hierarchical attentive recurrent tracking," *CoRR*, vol. abs/1706.09262, 2017. [Online]. Available: http://arxiv.org/abs/1706.09262

[6] A. Bewley, Z. Ge, L. Ott, F. Ramos, and B. Upcroft, "Simple online and realtime tracking," in *2016 IEEE International Conference on Image Processing (ICIP)*, 2016, pp. 3464–3468.

[7] R. Urtasun, D. J. Fleet, and P. Fua, "3d people tracking with gaussian process dynamical models," in *2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'06)*, vol. 1, June 2006, pp. 238–245.

[8] W. B. S. Thrun and D. Fox, *Probabilistic Robotics*. MIT Press, 2010.

[9] S. Scheidegger, J. Benjaminsson, E. Rosenberg, A. Krishnan, and K. Granstrm, "Mono-camera 3d multi-object tracking using deep learning detections and pmbm filtering," in *2018 IEEE Intelligent Vehicles Symposium (IV)*, June 2018, pp. 433–440.

[10] P. Z. Dan Iter, Jonathan Kuck, "Target tracking with kalman filtering, knn and lstms," December 2016.

[11] P. Chu and H. Ling, "Famnet: Joint learning of feature, affinity and multi-dimensional assignment for online multiple object tracking," *CoRR*, vol. abs/1904.04989, 2019. [Online]. Available: http://arxiv.org/abs/1904.04989

[12] N. Wojke and A. Bewley, "Deep cosine metric learning for person re-identification," in *2018 IEEE Winter Conference on Applications of Computer Vision (WACV)*. IEEE, 2018, pp. 748–756.

[13] A. Rangesh and M. M. Trivedi, "No blind spots: Full-surround multi-object tracking for autonomous vehicles using cameras and lidars," *IEEE Transactions on Intelligent Vehicles*, vol. 4, no. 4, pp. 588–599, December 2019.

[14] A. Asvadi, P. Peixoto, and U. Nunes, "Detection and tracking of moving objects using 2.5d motion grids," in *2015 IEEE 18th International Conference on Intelligent Transportation Systems*, September 2015, pp. 788–793.

[15] S. Song, Z. Xiang, and J. Liu, "Object tracking with 3d lidar via multi-task sparse learning," in *2015 IEEE International Conference on Mechatronics and Automation (ICMA)*, August 2015, pp. 2603–2608.

[16] C. R. Qi, H. Su, K. Mo, and L. J. Guibas, "Pointnet: Deep learning on point sets for 3d classification and segmentation," *arXiv preprint arXiv:1612.00593*, 2016.

[17] C. R. Qi, L. Yi, H. Su, and L. J. Guibas, "Pointnet++: Deep hierarchical feature learning on point sets in a metric space," *arXiv preprint arXiv:1706.02413*, 2017.

[18] H. Karunasekera, H. Wang, and H. Zhang, "Multiple object tracking with attention to appearance, structure, motion and size," *IEEE Access*, vol. 7, pp. 104 423–104 434, 2019.

[19] Y. Xiang, A. Alahi, and S. Savarese, "Learning to track: Online multi-object tracking by decision making," in *2015 IEEE International Conference on Computer Vision (ICCV)*, December 2015, pp. 4705–4713.

[20] P. Lenz, A. Geiger, and R. Urtasun, "Followme: Efficient online min-cost flow tracking with bounded memory and computation," July 2014.

[21] Y. Xu, S. Chen, Z. Wang, and L. Kang, "Modified joint probability data association algorithm controlling track coalescence," in *2011 Fourth International Conference on Intelligent Computation Technology and Automation*, vol. 1, March 2011, pp. 442–445.

[22] S. S. Blackman, "Multiple hypothesis tracking for multiple target tracking," *IEEE Aerospace and Electronic Systems Magazine*, vol. 19, no. 1, pp. 5–18, January 2004.

[23] R. G. Shaoqing Ren, Kaiming He and J. Sun, "Faster r-cnn: Towards real-time object detection with region proposal networks," January 2016.

[24] S. Shi, X. Wang, and H. Li, "Pointrcnn: 3d object proposal generation and detection from point cloud," in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019.

[25] A. H. Lang, S. Vora, H. Caesar, L. Zhou, J. Yang, and O. Beijbom, "Pointpillars: Fast encoders for object detection from point clouds," in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019.

[26] J. Ku, M. Mozifian, J. Lee, A. Harakeh, and S. L. Waslander, "Joint 3d proposal generation and object detection from view aggregation," in *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Oct 2018, pp. 1–8.

[27] C. R. Qi, W. Liu, C. Wu, H. Su, and L. J. Guibas, "Frustum pointnets for 3d object detection from rgb-d data," in *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, June 2018, pp. 918–927.

[28] W. Luo, B. Yang, and R. Urtasun, "Fast and furious: Real time end-to-end 3d detection, tracking and motion forecasting with a single convolutional net," June 2018, pp. 3569–3577.

[29] A. Paszke, S. Gross, S. Chintala, G. Chanan, E. Yang, Z. DeVito, Z. Lin, A. Desmaison, L. Antiga, and A. Lerer, "Automatic differentiation in pytorch," 2017. [Online]. Available: https://academic.microsoft.com/paper/2899771611

[30] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," 2014.

[31] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. C. Berg, and L. Fei-Fei, "ImageNet Large Scale Visual Recognition Challenge," *International Journal of Computer Vision (IJCV)*, vol. 115, no. 3, pp. 211–252, 2015.

[32] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay, "Scikit-learn: Machine learning in Python," *Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011.

[33] K. Bernardin and R. Stiefelhagen, "Evaluating multiple object tracking performance: The clear mot metrics," *EURASIP Journal on Image and Video Processing*, vol. 2008, January 2008.

[34] X. Weng and K. Kitani, "A baseline for 3d multi-object tracking," *CoRR*, vol. abs/1907.03961, 2019. [Online]. Available: http://arxiv.org/abs/1907.03961

[35] J. Snoek, H. Larochelle, and R. P. Adams, "Practical bayesian optimization of machine learning algorithms," 2012.