2018 IEEE Intelligent Vehicles Symposium (IV)
Changshu, Suzhou, China, June 26-30, 2018

A Lightweight Online Multiple Object Vehicle Tracking Method

Gültekin Gündüz[1] and Tankut Acarman[2]

Abstract— **In this paper, multiple-object vehicle tracking system by affinity matching using min-cost linear cost assignment is proposed. This tracking system is targeted to scene recordings acquired from cameras mounted on a moving ego vehicle. Vehicle tracking on the road scene and images acquired from moving ego vehicle's camera amplifies the problem of greater bounding box geometry change in comparison with other low speed tracking applications such as traditional pedestrian tracking. This perturbation occurs in many tracking scenarios such as when a high speed object is approaching from an opposing lane. Since autonomous driving algorithms need to use the processing resources in an efficient manner even while satisfying the requirements of computationally complex tasks like localization, object detection, occupancy grid update, sensor-fusion and trajectory planning, our study is particularly focused on the development and benchmarking of an computationally lightweight online multiple object tracking model. To test and evaluate our model, we use KITTI Object Tracking - Car Benchmark dataset and our model statistical metric values are comparably higher; our model outperforms the state-of-the-art methods on ML and MT and places second on MOTA and MOTP metric evaluations, and processing time is 6 to 20 times faster compared to other methods.**

I. Introduction

Multi object tracking with its prediction capability about surrounding dynamic traffic scene plays a crucial role in autonomous driving subject to safety-critical tasks such as trajectory planning and decision making [1], [2]. Performance enhancement in runtime and detection accuracy of Convolutional Neural Network (CNN) has created the "tracking-by-detection" paradigm [3]–[5]. A network delivering higher accurate and lower number of false negatives needs to embed more complexity with a high number of parameters to be tuned and more processing requirements [6]. Considering localization, object detection, sensor-fusion, occupancy grid update, trajectory planning, dynamical modelling and control alike tasks used in autonomous driving [7], [8], both computationally efficient and accurate solutions are needed for widespread usage. Multiple-Object trackers are divided whether LIDAR point clouds, stereo-pair image, single camera image sensors are used and whether an online or batch processing method is adopted. We present "extraCK" a lightweight online multiple object vehicle tracking method, which is an online "tracking-by-detection" multiple-object vehicle tracker that relies on a single camera.

[1] Gültekin Gündüz is with the Computer Engineering Department, Galatasaray University, 34349, Istanbul, Turkey ggunduz@sabanciuniv.edu

[2] Tankut Acarman is with the Computer Engineering Department, Galatasaray University, 34349, Istanbul, Turkey tacarman@gsu.edu.tr

II. Related Work

Online Multi-Object Tracking (MOT) has been widely studied. Due to the noisy detections, association with previously tracked objects is a challenging task. Markov Decision Process (MDP) has been adopted such as "birth/death" and "appearance/disappearance" of targets are treated as state transitions in the MDP [9]. Min-cost flow optimization for MOT has been studied in [10], and cost of the 'detection', 'birth-death' and 'transition between detection' edges are learned. An optimal set of tracks with quadratic interactions is extracted using greedy algorithms and linear programming. Appearance, motion and interaction features are encoded and combined using Long Short-Term Memory (LSTM) model in [11], promising results for pedestrian tracking were obtained. Single CNN based object tracker has been introduced, each target's specific CNN branch is discovered, online and extracted features are combined with targets motion model in [12]. Quadruplet CNN has been used in [13], tracklet assignments across frames are done according to quadruplet losses. Detected objects are gridded and local flow descriptors are binned according to their position, appearance similarity, target dynamics and trajectory regularization is combined and model is formulated as an energy minimization framework for the set of all hypotheses. Detection of objects and changing points has been studied in [14], by following [15] detections and point trajectories are defined as a graphical model, and minimum cost multi-cut problem of pairwise potentials are solved.

Target specific similarity functions have been studied, for a temporal local window object appearance similarity functions are learned online and min-cost multi-commodity flow problem is solved in [16]. Online target based appearance and motion cues learning by utilizing network flow optimization for tracklet affinity has been proposed in [17]. [18] has focused on complications of scenes acquired from cameras mounted on moving vehicles, which presents similar goal followed by our methodology. Structural motion constraints, which are described by location and velocity difference between objects, are compared to the detection anchors and assignment with the minimum cost is presented.

III. Method

A. Affinity Features

For each detected object, a variety of features can be extracted for affinity measurement such as geometric features containing bounding box coordinates, width, height, disparity measure depending on the availability of stereo images, and occlusion percentage by other detected objects. Appearance

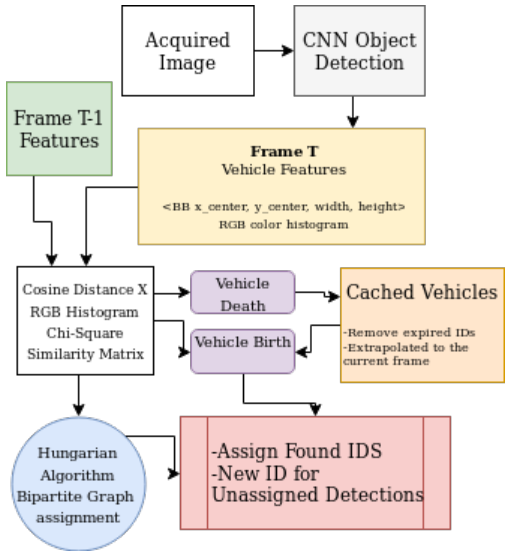

Fig. 1: Flowchart of the tracker model. Acquired image is processed by the CNN for object detection. For each detection feature vectors and RGB color histograms are compared with the previous frame's. Object IDS are assigned according to their affinity measurement.

based features include inspection of the color histograms, or keypoint descriptors [19]. Accurate measurement of pairwise affinity between detected objects of two consecutive frames is the key challenge for multiple-object tracking. Due to the mobile vision platform, a large number of motion scenarios can be witnessed like ego vehicle merging a road with right turn and a vehicle travels through the opposing lane.

In these circumstances, if given distance between the feature vectors is adequately large, a new ID is assigned otherwise the number of switching between assigned IDs increases in crowded situations. Overall, selection of generalized informative feature set for affinity measurement is a challenging task.

Dependent on tracking scenario and situation, specific problems can occur subject to different features. Keypoint descriptors can be unavailable for occluded objects or they can be located far away within small bounding box area. Using distance information seems intuitive but there may be some particular cases as highlighted in Figure 2 when an detected object which is partially occluded by the environment, various disparity measures can occur. In Figure 2, frames from the left camera images are given on the first row, and disparity map estimation by following [20] is plotted in the second row. Even for low rates of occlusion, since already minimum and maximum disparity measure do not give any useful knowledge, mean disparity measure exhibits inconsistencies. Advanced processing methods like occlusion-pose estimation [21], occlusion classifiers [22] or histogram comparisons are needed. Appearance of the detected object is shown to be informative, RGB channel histograms are generally binned to arbitrary number of bins [23]. Even for low number of bins if the values of the three channels are added to the feature vector, bounding box position is absorbed, both weighted distance calculation and learning of the weight parameters is required.

Lastly if the occlusion ratio of an object by the other detected objects is used, when the near-most object that occludes another disappears, values are exchanged between these two objects. Feature vector, F_i is defined by the specific values of its bounding box as follows:

$F_i =< x-center, y-center, width, height >$

where the width and height related feature is normalized between the range of 0 and 1 according to the size of the acquired frame. Also the 3-dimensional RGB histogram of the $i-th$ bounding box patch, denoted by H_i, is extracted with 6-bins for each channel. The RGB histogram is normalized and flatten into one dimension, resulting with the length of 216.

B. Tracker Model

Objects in an acquired image are detected using Faster R-CNN ([3]) with 300 region proposals and anchor stride lengths of 8 pixels, backbone convolutional network is a pretrained Inception-Resnet-V2 model that is pre-trained on ImageNet [24] dataset which is finetuned using KITTI 2D object detection dataset [4], [5]. For each detected object affinity features, denoted by F_i, are extracted and for frame $t > 1$, pairwise feature cosine distance matrix $D_{i\times j}$ is derived as follows:

$$D_{i,j} = 1 - \frac{i \cdot j}{||i||_2 \cdot ||j||_2}, i \in F_t, j \in F_{t-1} \quad (1)$$

Also histograms are compared using Chi-squared distance where $S_{i\times j}$ represents the chi-squared distance matrix of RGB color histograms. Again for $t > 1$, chi-squared distance of two histograms is given by:

$$S_{i,j} = \chi^2(H_i, H_j) = \sum_I \frac{(H_i(I) - H_j(I))^2}{H_i(I)} \quad (2)$$

when pairwise feature distances and histogram similarity is assigned, affinity cost matrix $C_{i\times j}$ is calculated by:

$$C_{i,j} = D_{i,j} S_{i,j} \quad (3)$$

once the cost matrix is established, row and column minimums are extracted in order to determine whether the previously tracked object is disappeared or a new object is appeared. In such a case, an affinity cost is computed greater than determined threshold, disappeared vehicles are cached and then new objects are cross checked and compared versus the cached ones. When similarity is below a certain threshold, then the cached ID is re-assigned otherwise a new ID is created. Figure 3 shows the feature distance matrix, chi-squared histogram distance and the cost matrix from top to bottom, respectively. Both the disappeared vehicle columns and new appeared rows are removed from the cost matrix.

Remaining vehicles present in the cost matrix are assigned solving a linear sum assignment problem, minimum weight

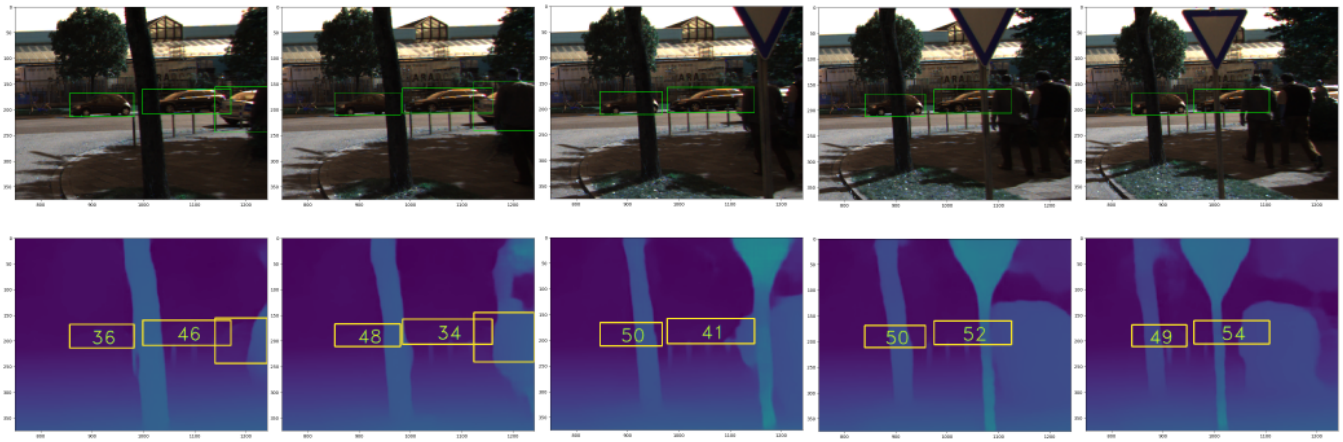

Fig. 2: Top row shows detected vehicles and bottom row stereo disparity map for KITTI Object Tracking Training Sequence 0001, Frames 160 through 164. Mean disparity values of the area marked by the bounding box is shown in yellow. Even low rates of occlusion causes mean disparity measure of the detected object to fluctuate through the sequence, and does not provide a stable affinity feature.

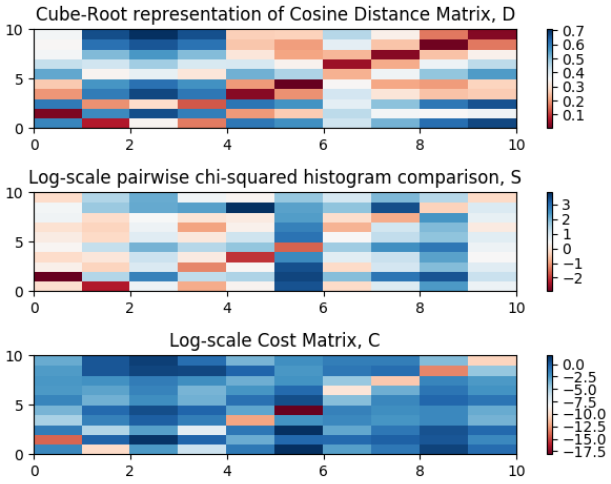

Fig. 3: Feature cosine distance matrix, chi-squared RGB color histogram similarity and cost matrix is represented from top to bottom respectively. Row indexes represents detections of objects at frame t and columns indexes those of the previous frame, $t-1$.

matching of a bipartite graph introduced by the Hungarian Algorithm [25]. If X is a boolean matrix and $X(i,j) = 1$ if and only if row i is assigned to the column j, optimal assignment is determined by solving:

$$min \sum_i \sum_j C_{i,j} X_{i,j} \tag{4}$$

Assignment is done for the square matrix of order $min(i,j)$, so if an object is not assigned to any previous detection or not determined similar to the cached objects, a new tracklet is assigned.

C. Post Frame Processing

When the tracklet assignments of the current frame are completed both the cached and active vehicles features are predicted for the next frame. This task serves three purposes: adjustment of the cost matrix, identification of a previously tracked object that has disappeared but reappearing in the following frames, and adjustment of bounding box behaviour for near objects, which may also displace with high velocity. The object features are predicted for the frame at time index $t+1$ by applying the least squares method to fit the line at time index t. For all objects that are either active or cached, if data have been provided for number of frames higher than the given threshold, fitted value at $t+1$ for each of the features is extracted. Extrapolated feature value is replaced by the observed one and used for feature vector distance comparison in the next frame.

In Figure 5, a tracked object is approaching from the opposite direction with respect to the ego vehicle. Due to the high relative speed between the camera and detected object, bounding box features show considerable change. If such considerable feature distance is accepted by the affinity model, the prediction performance can be significantly degraded in crowded tracking scenarios. However, extrapolating the feature vector toward the following frame gives insight about the next possible bounding box and also whether if vehicle is only partially visible. If the bounding box is extrapolated outside the frame limits, the part expected to be out of the frame is excluded. The frame placed in the bottom left of Figure 5 shows the cosine distances between bounding box of the detection at frame 97 and predicted bounding box for frame 97 and observed bounding box in frame 96.

Also a previously tracked vehicle can not be observed due to occlusions or false negatives by the CNN. An illustrative example is shown at Figure 4. Same vehicle is last detected by the CNN at frame 73, which is plotted in the rightmost

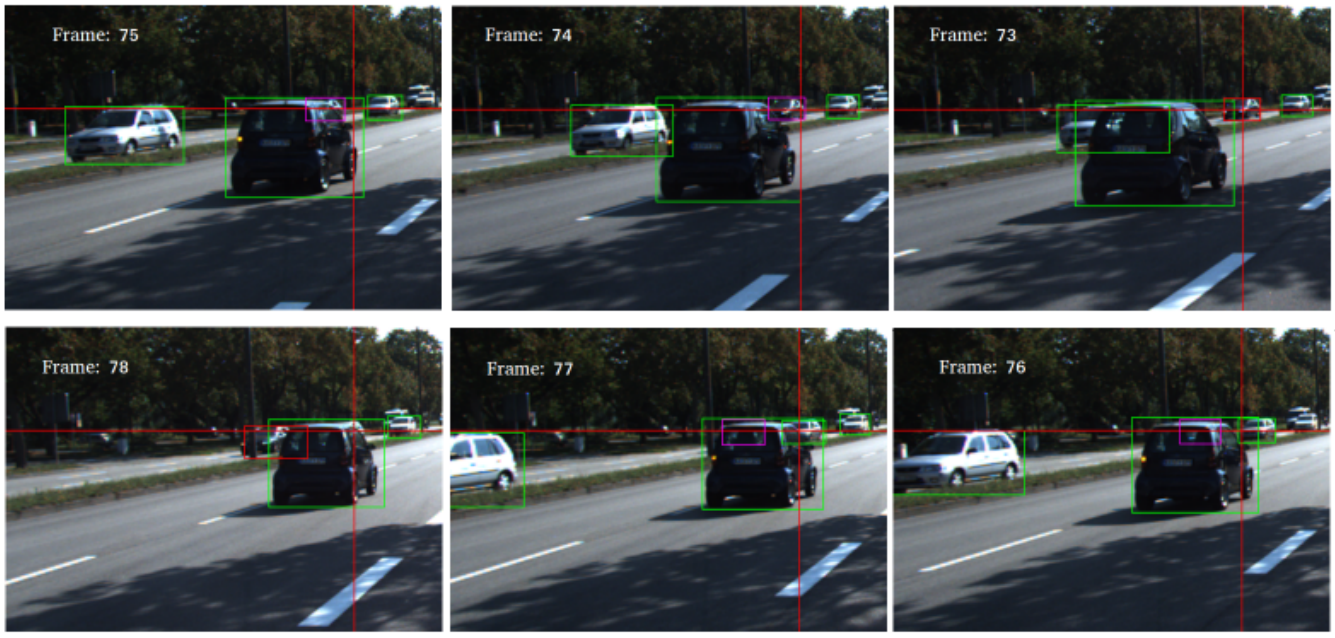

Fig. 4: KITTI Object Tracking Testing Sequence 0007, Frames 73 through 78. Top right image shows the last detection of a vehicle, where red horizontal and vertical lines represents the bounding box coordinates. For frames 74-77 purple bounding box is the predicted movement of the same vehicle. Left most figure, Frame 78 shows the bounding box when the same vehicle is detected.

Fig. 5: The sequence of frames numbered 94 through 97 belonging to Tracking Training Sequence 0008 represents considerable bounding box size changes between consecutive frames. For affinity matching, bounding box of a tracked object that is expected to be partially visible the following frame is adjusted.

of the first row, and is not re-detected until frame 78, plotted in the leftmost of the second row. During this period of disappearance, the cached vehicles features are extrapolated for minimizing the feature distance on reappearing.

IV. Evaluation

For evaluation purposes, KITTI Object Tracking Evaluation 2012 dataset in [2] is used and only the 'Car' class is considered. Training dataset consists of 21 sequences with 8.008 frames and testing dataset consists of 29 sequences with 11.095 frames. Frames were recorded at 10 FPS from a camera mounted on a ego vehicle. All sequences have varying number of objects and lengths with their unique motion scenarios. In our evaluation study, the following metrics are adopted: CLEAR MOT [26] and also Fragmentation(FRAG), ID-switch (IDS), Mostly-Tracked (MT) and Mostly-Lost (ML), which are defined in [27].

Table I presents the statistical metric values of Recall, Precision, F-measure, False Alarm Rate (FAR) rates and number of True Positive (TP), False Positive (FP), False Negative (FN), False Alarm Rate (FAR) of our method "extraCK". These metric values are the result of the object detection part of the tracker. Table II shows multi-object tracking related statistical metric values, namely Multiple Object Tracking Precision (MOTP) illustrates the ability of the tracker to estimate precise object positions, Multiple Object Tracking Accuracy (MOTA) is the ratio of the total sum of FN, FP and mismatches computed over the total number of frames versus the total number of ground truth objects, [26]. MT is defined as the percentage of output trajectories that cover more than 80% of ground truth trajectories, ML is the percentage of output trajectories that cover less than 20% of the ground truth trajectories, IDS is the number of times a tracked trajectory is changed and FRAG defines the number of times a ground truth trajectory is interrupted. The performance of our method is compared with the state-of-the-art methods tested subject to 'Car' class of KITTI Tracking Benchmark (see for instance, [2]), the list benchmarking object tracking evaluation methods versus different

TABLE I: 'Car' class detection metrics comparison

Method	Recall	Precision	F1	TP	FP	FN	FAR
IMMDP [9]	86.11 %	98.82 %	92.03 %	32668	391	5269	3.51 %
MCMOT-CPD [14]	81.84 %	98.87 %	89.59 %	30247	316	6713	2.84 %
NOMT* [19]	83.22 %	96.78 %	89.49 %	31854	1061	6421	9.54 %
LP-SSVM* [10]	83.35 %	96.27 %	89.34 %	31997	1239	6393	11.14 %
MDP [11]	80.26 %	98.00 %	88.25 %	29747	606	7315	5.45 %
SCEA* [18]	81.76 %	96.00 %	88.31 %	31330	1306	6989	11.74 %
extraCK	84.51 %	98.04 %	90.77 %	32156	642	5896	5.77 %

TABLE II: MOT Metrics Comparison

Method	MOTA	MOTP	MT	ML	IDS	FRAG	Runtime
IMMDP [9]	83.04 %	82.74 %	60.62 %	11.38 %	172	365	0.19 s
MCMOT-CPD [14]	78.90 %	82.13 %	52.31 %	11.69 %	228	536	0.01 s
NOMT* [19]	78.15 %	79.46 %	57.23 %	13.23 %	31	207	0.09 s
LP-SSVM* [10]	77.63 %	77.80 %	56.31 %	8.46 %	62	539	0.02 s
MDP [11]	76.59 %	82.10 %	52.15 %	13.38 %	130	387	0.9 s
SCEA* [18]	75.58 %	79.39 %	53.08 %	11.54 %	104	448	0.06 s
extraCK	79.99 %	82.46 %	62.15 %	5.54 %	343	938	0.03 s

metric values is available at `http://www.cvlibs.net/datasets/kitti/eval_tracking.php`. At the time of writing this paper, our model ranks second on the MOTA metric because of its higher number of FN metric value and IDS value in comparison with the better performing method, see for instance the seven methods in Table II. Performance of locating the tracked object, i.e., the MOTP value, is again ranked second and relatively close to the first ranked method. To capture the majority of the tracklet, according to MT statistical metric value and also in terms of ML statistical metric value, our method "extraCK" outperforms the other state-of-the-art methods. Benchmarking performance of our method is visualized and compared versus the state-of-the-art methods in Figure 6.

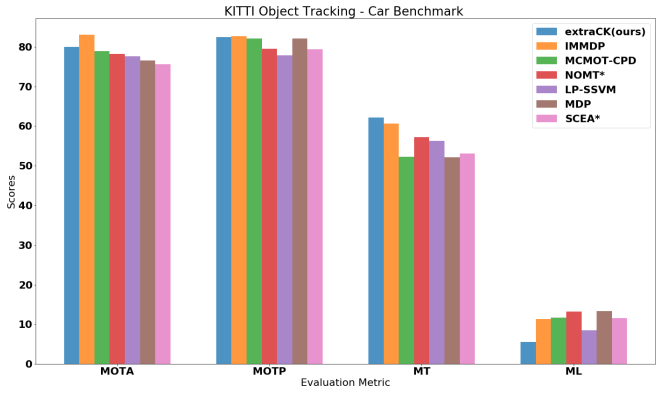

Fig. 6: 'Car' class multiple object tracking accuracy (MOTA), multiple object tracking precision (MOTP), mostly tracked (MT) and mostly lost (MT) metric comparison to other published methods in the KITTI Object Tracking Benchmark.

Computational complexity of tracklet assignment with affinity cost matrix depends on the minimum of the number of current and previous detections, detailed analysis is presented in [25], [28]. Considering the runtime performance while being tested with the frames involving an detected object on an Intel i7-6820HK at 2.70 GHz CPU, our tracker model is performed subject to a mean running time of 0.0295 seconds or $\approx 34FPS$, having a standard deviation of 0.01228 seconds. Runtime histogram and probability distribution is plotted in Figure 7.

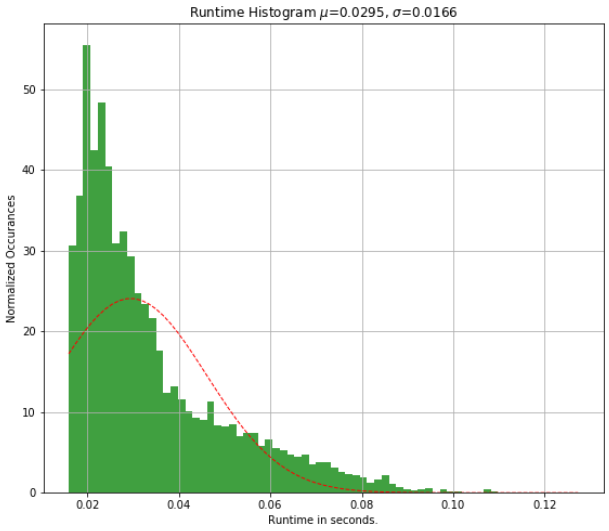

Fig. 7: Runtime histogram and probability distribution plot, determined by the minimum of the number of current and previously detected objects. Model works with a mean $\approx 34Hz$ on KITTI Object Tracking sequences.

V. Conclusion

A lightweight online multiple object vehicle tracking method "extraCK" solving the min-cost linear sum assignment problem of extrapolated motion combined with appearance features is presented. Considering ML metric, our extraCK method outperforms the state-of-the-art methods, while also our method is benchmarked in the top three metric results for MOTA, MOTP and MT tested subject to the 'Car' class of KITTI Object Tracking Benchmark. The runtime performance, which is 0.03 seconds, is tested and cross-checked. A computationally 'lightweight' multiple vehicle tracking for autonomous driving is achieved with a speed increase over 6 to 20 times versus the other methods. The achieved runtime performance enables trajectory planning in accordance to the motion of the surrounding vehicles in the challenging tracking scenes. Runtime performance along the acceptable level of tracking metric values enables computational resources of autonomous vehicles to be used by other time-critical tasks.

VI. Acknowledgements

The authors gratefully acknowledge the support of Galatasaray University, scientific research support program under grant # 17.401.003.

References

[1] A. Petrovskaya and S. Thrun, "Model based vehicle detection and tracking for autonomous urban driving," *Autonomous Robots*, vol. 26, no. 2-3, pp. 123–139, 2009.

[2] A. Geiger, P. Lenz, and R. Urtasun, "Are we ready for autonomous driving? the kitti vision benchmark suite," in *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*, pp. 3354–3361, IEEE, 2012.

[3] S. Ren, K. He, R. Girshick, and J. Sun, "Faster r-cnn: Towards real-time object detection with region proposal networks," in *Advances in neural information processing systems*, pp. 91–99, 2015.

[4] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770–778, 2016.

[5] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, "Rethinking the inception architecture for computer vision," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2818–2826, 2016.

[6] J. Huang, V. Rathod, C. Sun, M. Zhu, A. Korattikara, A. Fathi, I. Fischer, Z. Wojna, Y. Song, S. Guadarrama, *et al.*, "Speed/accuracy trade-offs for modern convolutional object detectors," *arXiv preprint arXiv:1611.10012*, 2016.

[7] C. Urmson, J. Anhalt, D. Bagnell, C. Baker, R. Bittner, M. Clark, J. Dolan, D. Duggins, T. Galatali, C. Geyer, *et al.*, "Autonomous driving in urban environments: Boss and the urban challenge," *Journal of Field Robotics*, vol. 25, no. 8, pp. 425–466, 2008.

[8] J. Levinson, J. Askeland, J. Becker, J. Dolson, D. Held, S. Kammel, J. Z. Kolter, D. Langer, O. Pink, V. Pratt, *et al.*, "Towards fully autonomous driving: Systems and algorithms," in *Intelligent Vehicles Symposium (IV), 2011 IEEE*, pp. 163–168, IEEE, 2011.

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

[10] S. Wang and C. C. Fowlkes, "Learning optimal parameters for multi-target tracking with contextual interactions," *International Journal of Computer Vision*, vol. 122, no. 3, pp. 484–501, 2017.

[11] A. Sadeghian, A. Alahi, and S. Savarese, "Tracking the untrackable: Learning to track multiple cues with long-term dependencies," *arXiv preprint arXiv:1701.01909*, 2017.

[12] Q. Chu, W. Ouyang, H. Li, X. Wang, B. Liu, and N. Yu, "Online multi-object tracking using cnn-based single object tracker with spatial-temporal attention mechanism," *arXiv preprint arXiv:1708.02843*, 2017.

[13] J. Son, M. Baek, M. Cho, and B. Han, "Multi-object tracking with quadruplet convolutional neural networks," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 5620–5629, 2017.

[14] B. Lee, E. Erdenee, S. Jin, M. Y. Nam, Y. G. Jung, and P. K. Rhee, "Multi-class multi-object tracking using changing point detection," in *European Conference on Computer Vision*, pp. 68–83, Springer, 2016.

[15] M. Keuper, S. Tang, Y. Zhongjie, B. Andres, T. Brox, and B. Schiele, "A multi-cut formulation for joint segmentation and tracking of multiple objects," *arXiv preprint arXiv:1607.06317*, 2016.

[16] M. Yang, Y. Wu, and Y. Jia, "A hybrid data association framework for robust online multi-object tracking," *arXiv preprint arXiv:1703.10764*, 2017.

[17] B. Wang, G. Wang, K. L. Chan, and L. Wang, "Tracklet association by online target-specific metric learning and coherent dynamics estimation," *IEEE transactions on pattern analysis and machine intelligence*, vol. 39, no. 3, pp. 589–602, 2017.

[18] J. Hong Yoon, C.-R. Lee, M.-H. Yang, and K.-J. Yoon, "Online multi-object tracking via structural constraint event aggregation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1392–1400, 2016.

[19] W. Choi, "Near-online multi-target tracking with aggregated local flow descriptor," in *Proceedings of the IEEE International Conference on Computer Vision*, pp. 3029–3037, 2015.

[20] N. Mayer, E. Ilg, P. Häusser, P. Fischer, D. Cremers, A. Dosovitskiy, and T. Brox, "A large dataset to train convolutional networks for disparity, optical flow, and scene flow estimation," in *IEEE International Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016. arXiv:1512.02134.

[21] C. Wang, Y. Fang, H. Zhao, C. Guo, S. Mita, and H. Zha, "Probabilistic inference for occluded and multiview on-road vehicle detection," *IEEE Transactions on Intelligent Transportation Systems*, vol. 17, no. 1, pp. 215–229, 2016.

[22] M. Mathias, R. Benenson, R. Timofte, and L. Van Gool, "Handling occlusions with franken-classifiers," in *Proceedings of the IEEE International Conference on Computer Vision*, pp. 1505–1512, 2013.

[23] K. Meshgi and S. Ishii, "Expanding histogram of colors with gridding to improve tracking accuracy," in *Machine Vision Applications (MVA), 2015 14th IAPR International Conference on*, pp. 475–479, IEEE, 2015.

[24] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "Imagenet: A large-scale hierarchical image database," in *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*, pp. 248–255, IEEE, 2009.

[25] H. W. Kuhn, "The hungarian method for the assignment problem," *Naval Research Logistics (NRL)*, vol. 2, no. 1-2, pp. 83–97, 1955.

[26] K. Bernardin and R. Stiefelhagen, "Evaluating multiple object tracking performance: the clear mot metrics," *EURASIP Journal on Image and Video Processing*, vol. 2008, no. 1, p. 246309, 2008.

[27] Y. Li, C. Huang, and R. Nevatia, "Learning to associate: Hybrid-boosted multi-target tracker for crowded scene," in *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*, pp. 2953–2960, IEEE, 2009.

[28] J. Munkres, "Algorithms for the assignment and transportation problems," *Journal of the society for industrial and applied mathematics*, vol. 5, no. 1, pp. 32–38, 1957.