MULTI-OBJECT TRACKING BY ASSOCIATIONS ON TEMPORAL WINDOW

(GECICI PENCEREDE CAGRISIMLARA DAYALI COKLU NESNE TAKIBI)

by

GULTEKIN GUNDUZ, B.S.

Thesis

Submitted in Partial Fulfillment

of the Requirements

for the Degree of

MASTER OF SCIENCE

in

COMPUTER ENGINEERING

in the

GRADUATE SCHOOL OF SCIENCE AND ENGINEERING

of

GALATASARAY UNIVERSITY

JUNE 2018

This is to certfiy that the thesis entitled

MULTI-OBJECT TRACKING BY ASSOCIATIONS ON TEMPORAL WINDOW

prepared by **GULTEKIN GUNDUZ** in partial fulfillment of the requirements for the degree of **Master of Science in Computer Engineering Department** at **Galatasaray University** approved by the

Examining Committee Members:

Prof. Dr. Tankut ACARMAN
Supervisor, **Computer Engineering Department, GSU** _____

Prof. Dr. Mehmet Turan SÖYLEMEZ
Control and Automation Engineering Department, ITU _____

Dr. Öğr. Üyesi Murat AKIN
Computer Engineering Department, GSU _____

Date: _____

ACKNOWLEDGMENTS

I would like to thank the members of Galatasaray University for the time I spent during my Master of Science study, especially Prof. Dr. Tankut Acarman.

June 2018
Gültekin GÜNDÜZ

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

ABSTRACT

In this thesis, multiple-object vehicle tracking system that generates tracklets by solving the min-cost max flow problem of affinity between detected objects of consecutive frames are proposed. Recent performance enhancement in runtime and detection accuracy of Convolutional Neural Networks and their capability is object detection created the *tracking-by-detection* paradigm.

Tracking-by-detection is the approach of data i.e. identity, association between individual frames of a sequence. The proposed tracking system is targeted to autonomous driving applications, with being able to track on non-stationary scene recordings, i.e. from a moving ego vehicle. Object detection task is performed using an Neural Network Ensemble consisting of 3 Faster R-CNN Inception ResNet v2 networks, that are trained on ImageNet dataset and fine-tuned for KITTI Object Detection Dataset.

A method is proposed for combining the bounding boxes generated from each of the Convolutional Neural Network. Efficient usage of the processing resources for autonomous driving that should satisfying the requirements of computationally complex tasks such as localization, object detection, occupancy grid update, sensor-fusion, trajectory planning etc. are respected by generating tracklets on a temporal window of three frames.

Also data association is done by solving min-cost flow of sparse affinity network rather than considering all possible assignments, that solves memory and computational constraints. Lighter version of the method is also presented where the temporal window is reduced to two consecutive frames and data association is done by solving linear sum assignment problem, minimum weight matching of a bipartite graph introduced by the Hungarian Algorithm. For evaluation of the trackers KITTI Object Tracking Evaluation 2012 dataset and 'Car' class is used. KITTI Object Tracking dataset consists of 21 training sequences with 8.008 frames and 29 testing sequences with 11.095 frames. Frames were recorded at 10 FPS from a camera mounted on the 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: Recall, Preci-

sion, F1-Measure, False Alarm Rate, False Positives, False Negatives relating to the object detection task and Runtime, widely used CLEAR MOT metrics like Multiple Object Tracking Accuracy (MOTA) and also Fragmentation (FRAG),ID-switch (IDS), Mostly-Tracked (MT) and Mostly-Lost (ML) for MOT evaluation. with our model performing second on MOTA, MT, ML metrics compared to the state-of-the-art online MOT methods. It showed less than half of the reported IDS from the best MOTA metric and lower FRAG, while working 6 times faster, with mean runtime at 20 Hz.

Keywords : object detection, graph theory, multiple object tracking, intelligent transportation systems

ÖZET

Bu tezde görüntü üzerinde çoklu nesne araç takip sistemi önerilmiştir. Ardışık çerçevelerde nesnenin birim pozisyon değişikliğini nesnelerin aidiyetleri uzerine minimum maliyet maksimum akış problemini çözerek oluşturulmaktadır. Son yıllardaki Evrişimli Sinir Ağları işleyiş süresi ve nesne tespiti performans artışları "tespit-ederek-takip" yaklaşımını oluşturmuştur. "Tespit-ederek-takip" yaklaşımı ardışık çerçevelerde tespit edilmiş nesneler uzerinden data veya kimlik cağrışımları yapılması yöntemidir. Onerilen takip sistemi otonom surus uygulamalarını hedef almaktadir; hareketli şekilde kaydedilmiş sahne kayıtları, hareket eden merkez araçtan, üzerinde çalışma kabiliyeti bulunmaktadir. Nesne tanımlama görevi 3 adet ImageNet veri seti üzerinde eğitilmiş daha sonrasında ise KITTI nesne tanimlama veri seti uzerinde son katman ayarları yapılmıs Faster R-CNN Inception ResNet v2 ağından oluşan Sinir Ağları Topluluğu ile gerçekleştirilmektedir. Ayrıca Sinir Ağları Topluluğunun ürettiği sınırlayıcı nesne pozisyon kutularini birşeltirilmesi icin yeni bir metod önerilmiştir. Otonom sürüş için gerekli sayısal olarak karmaşık gereksinimler, konumlandırma, doluluk kafesleme guncellemesi, alıcı füzyonu, yörünge planlama vb., gereksinimlerle onem duyularak hesaplama kaynakları iki veya üç görüntü çerçevesinde geçici zaman aralığına odaklanılarak efektif bir sekilde kullanılmıştır. Geçici penceredeki olası bütün cağrışımlar yerine seyrek yakınlık ağı oluşturularak minimum maliyet maksimum akış problemi çözülmüştür. Hesaplama bakımından daha hafif bir versiyonda denenmiştir, ardışık iki görüntü çerçevesinde tespit edilmiş nesneler için ikili graf benzerlik ağırlıkları doğrusal toplam atama problemi, Macar Algorithmasi ile çözülmüştür. Önerilen yöntemlerin değerlendirilmesi KITTI Nesne Takibi Değerlendirme 2012 veri setinde 'Araba' sınıfı ile gerçekleştirilmiştir. KITTI Nesne Takibi veri seti 8.008 görüntü çerçevesinden oluşan 21 eğitim ve 11.095 görüntü çerçevesinden oluşan 29 test sekansı içermektedir. Çerçeveler saniyede 10 defa ile ölçümlenip kaydedilmiş, her sekansin birbirinden farklı uzunluğu ve değerlendirdiği hareket senaryoları içermektedir. Önerilen model ortalama olarak 20 hertz frekans ile çalışmakta olup cevrimiçi modeller ile kıyalandığında KITTI siralamasinda MOTA, MT ve ML metriklerinde ikinci olup, IDS metriğinde en iyi performansı göstermiştir.

Anahtar Kelimeler : nesne tanima, graf teorisi, coklu nesne takibi, akilli ulasim sistemleri

1 INTRODUCTION

It has been stated that multiple 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 (Petrovskaya and Thrun, 2009; Geiger et al., 2012; Urmson et al., 2008). Due to the image classification and object detection results attained by Convolutional Neural Networks (CNN) (Krizhevsky et al., 2012), and more recently their performance enhancement in runtime and detection accuracy has created the *tracking-by-detection* paradigm (Ren et al., 2015; He et al., 2016; Szegedy et al., 2016). A network delivering higher accurate proposals and lower number of false negatives needs to embed more complexity with a high number of floating point operations per seconds according to the number of parameters of the backbone network, increased number of convolutional layers or lower strides and smaller sized filters, with greater processing requirements (Huang et al., 2016). Considering localization, object detections, sensor-fusion, occupancy grid update, trajectory planning, dynamical modelling and control alike tasks used in modern autonomous driving applications (Urmson et al., 2008; Levinson et al., 2011), both computationally efficient and accurate solutions are required due to the safety critical nature of autonomous driving. Multiple-Object Tracking methods are divided according to the type of sensors (Darms et al., 2008) whether RADAR, LIDAR point clouds, stereo-pair image, monocular camera image are used and whether an online, i.e. incoming stream or data, or batch processing approach is adopted.

In this thesis an computationally efficient online multiple object vehicle tracking method regarding the memory and processing constraints, that focuses on a temporal window of three consecutive frames is presented. Tracklets of the detected objects at time t_i for the temporal window are generated by solving min-cost max flow problem of a sparse network with limited number of edges that are created due to strong detection affinities from times t_{i-1} and t_{i-2}. Figure 1.1 shows a general block diagram representation of the proposed method.

Rest of the work is organized as follows : Firstly literature review on MOT applications are given in, followed by the object proposal extraction from convolutional neural net-

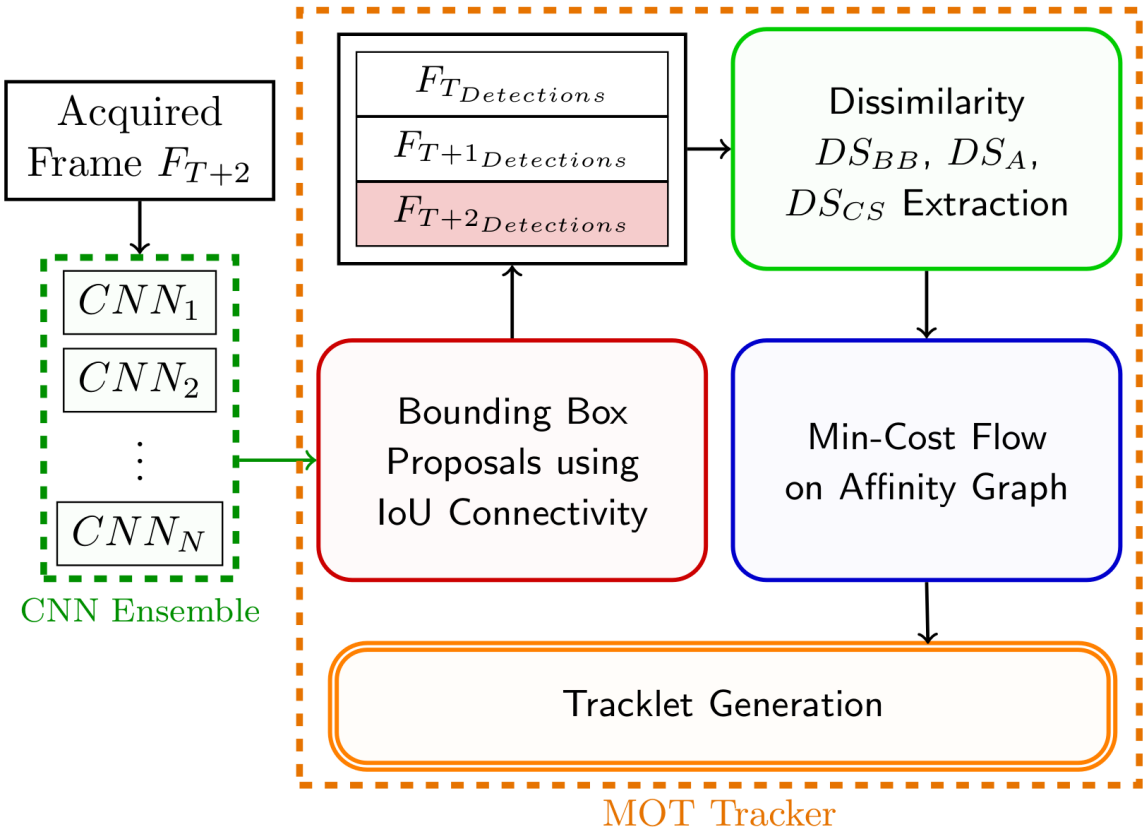

Figure 1.1: Block diagram of the proposed Multiple Object Tracking method. With low detection threshold object proposals are extracted from the convolutional neural network Ensemble for the current frame. Ensemble detections are filtered using intersection over union connectivity approach, which resulting detections dissimilarity are extracted with the previous two frames. Tracklet Generation is achieved by solving min-cost max flow of the affinity graph.

work Ensemble of the proposed method. Data association by affinity measurements of detected objects in the temporal window are explained. That is followed by explanation of the tracklet generation solving the min-cost max flow is explained and reports on experimental evaluation on KITTI Tracking Benchmark and runtime analysis. Finally, some conclusions are given.

2 LITERATURE REVIEW

Online Multi-Object Tracking (MOT) has been widely studied, with most of the methods adopting *tracking-by-detection* paradigm. An ensemble of convolutional neural network based object detection and Lucas-Kanede Tracker based motion detector is employed in (Lee et al., 2016) to compute the likelihood of foreground regions as the detection responses. Where the detections are assigned to tracklets according to their position changes that are formulated by a Bayesian filtering framework. Pairwise costs for objects tracks using 3D cues including object pose, shape and motion are used in (Sharma et al., 2018). Where it reported state-of-the-art results by bipartite matching the detected objects of two consecutive frames using Hungarian algorithm. In (Choi, 2015) affinity measure to associate detections are done using extracted keypoints from the image. Detections bounding boxes are grided and location of the keypoints are used for affinity measure which is followed by a "near-online" hypotheses generation. Tracklet assignment is done using Successive Shortest-Path in (Lenz et al., 2015), which is solved using a leveraged Dijkstra's algorithm, that focuses on the fact that only a small part of the graph is changed. Conditional Random Field is used in (Osep et al., 2017) to score the pairwise potential of each pair of hypotheses. Association affinity is linear combination of appearance score, motion models and projection model of observations. Minimum cost multi-cut formulation has been used in (Keuper et al., 2016)

Min-cost Flow has been used in (Wang and Fowlkes, 2017) for Multiple Object Tracking tracking. Collection of tracks that maximizes the posterior probability (MAP) are defined as transitions between objects of successive frames. Markov Decision Process has been used in (Xiang et al., 2015) with similarity between detections are encoding using optical flow quality, bounding box height ratio, bounding box overlap, detection score and euclidean distance between centers. Global camera movements has been studied in (Hong Yoon et al., 2016), where structural movement and position constraints are used to analyze detections relating to assigned anchors. Assignment is done according the minimum aggregated cost of the structural constraints. Online similarity learning is performed in (Yang et al., 2017) with local temporal window associations solving min-cost flow. Target specific metric learning and min-cost flow is also applied in (Wang et al., 2017) with ability to recover missed detections.

Tracking related to convolutional neural network properties has been also studied. Features from convolutional layers has been used for calculating affinity probabilities of detected objects in (Chen et al., 2017; Chu et al., 2017). Quadruplet convolutional neural network are used in that inputs frames from a temporal window of four frames (Son et al., 2017), which quadruplet loss enforcing temporally adjacent detections more closely located than the ones with large temporal gap. For a temporal window bounding boxes of the compared objects are input in CNNs and affinity is established by output feature vectors are examined using Long Short-Term Memory networks in (Sadeghian et al., 2017).

3 NEURAL NETWORK ENSEMBLE

Tracking-by detection approach detects independent objects in each frame than an affinity measure, i.e. data association, step is performed. Presence of occlusions, noisy detections consisting of false negatives and/or false positives makes the association problem difficult. A Multiple Object Tracking application is expected to perform better with increased detection recall and precision. Trade off between recall and precision has been widely studied (Rothe et al., 2014; Szegedy et al., 2014; Sun et al., 2012). As the threshold of an object proposal to be accepted decreases, a precision trade off occurs. A high recall value on the test data can be achieved by lower detection threshold, but would result with higher number of false positives. Additionally split of the convolutional neural network training data effects generalization performance. In our work to increase generalization we deployed an ensemble of three Faster R-CNN (Ren et al., 2015) with Inception-ResNet-v2 (Szegedy et al., 2016; He et al., 2016) backbones that has been fine tuned on KITTI Object Tracking dataset using a 0.8-0.2 train and validation splits. With an aim of proposing a bounding box selection method that maximises both recall and precision. Non-Maximum Suppression (NMS) (Girshick et al., 2014; Ren et al., 2015; Redmon et al., 2016; Hosang et al., 2017)is an integral part of detection approaches that ideally outputs a single bounding-box for each detection, if present merging multiple bounding boxes for an object. Challenges with NMS has been studied in (Rothe et al., 2014) where it can be summarized as the proposal with the highest confidence may not be the best fit, that it might suppress nearby object and lastly it does not provides insight about false positives. KITTI dataset consists of crowded scenes with high bounding box overlap.

Clustering methods can be an alternative for NMS bounding box combination, but challenges like whether to use clustering methods with even or uneven cluster size and deciding on the number of clusters emerge. Choosing the optimal number of clusters have been widely studied, like 'Elbow Method' (Tibshirani et al., 2001) which increments number of clusters until cluster intertia decrease reaches a platoon and examining Silhoutte Score of a cluster (Rousseeuw, 1987). However these approaches bring computational expenses by means of iterating over a range of possible clusters and does not perform well due to low number of data points and to the fact that only

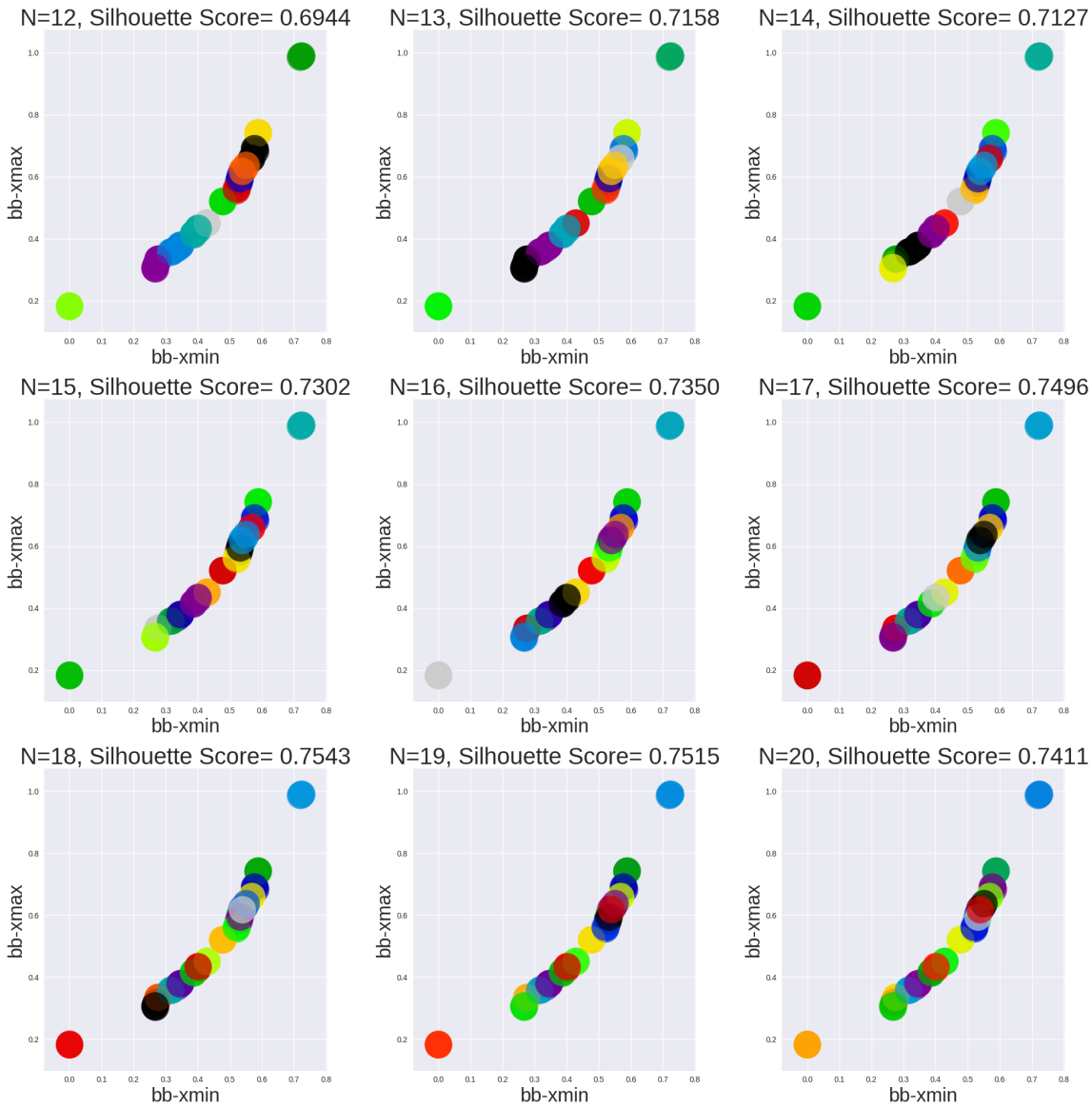

Figure 3.1: Finding optimal number of clusters relating to Silhoutte score. N represents the number of clusters and distinct colors represents clusters. Optimal cluster number is achieved for $N = 18$.

two dimensional position of the object is known, i.e. not available depth information. Clustering becomes problematic relating to scenes with high number of distant objects.

We present a method for bounding box combination from multiple convolutional neural network detections alternative to detection confidence based methods like NMS or clustering approaches. Our method favours proposal bounding box agreement of CNNs over detection confidence, for each detection over threshold t_{d_1} Intersection over Union (IoU) metric is extracted for the remaining detections. Any intersection over union smaller than the threshold t_{IoU} is set to zero and an intersection over union connectivity graph between proposed bounding boxes are acquired. Next step is to find the number of connected components of the graph. If any component consists of a single detection than it is only accepted if the detection confidence is higher than threshold t_{d_2}. For each found component mean values of the bounding box coordinates it is formed of is taken to produce a single detection. In 3.2 from top to bottom, all object proposals from the convolutional neural network Ensemble, ground truth kitti labels for classes 'Car' and 'DontCare' and extracted proposals from our method for KITTI Tracking Dataset, training sequence 11 frame 154 can be seen. Figure 3.1 show cluster analysis using Silhoutte Score for the same frame, that reports optimal number of clusters as 18 while our method produces 16 proposals, with less computation.

Figure 3.2: Top row shows all 'Car' class detections with confidence higher than 0.05 from convolutional neural network ensemble. Middle row represents the ground-truth bounding boxes provided by the KITTI dataset. Third row is the resulting object bounding boxes from the proposed method.)

4 AFFINITY MEASUREMENTS

Data association of detections from $frame_t$ and $frame_{t+1}$ is established using affinity measurements extracted from bounding box geometric, appearance based and relating to the changing scene properties. Each affinity property serves useful in specific scenarios and the aim is to establish if available a strong affinity by weighing according to the agreement of them all. For dense regions with multiple object proposals that are both far away from the camera and occlude each other, bounding box properties show similarity. In such situation however inspecting the color distribution of the detection produces more knowledge. While for a nearby object with relatively high width, as the object get occluded as a results the color distribution of the bounding box changes but its bounding box properties establish the strong affinity. Last affinity measurement used for data association is objects 's position relating to the changing scene. For a dynamic environment an objects position vector to the common keypoints detected is expected to be similar for the examined short amount of time, that strengths bounding box position similarity in case of high velocity objects or camera position changes. Situation specific problems are faced with some of the features. Keypoint descriptors can be unavailable for occluded objects, or that are far away with small bounding box area. Using distance information looks intuitive but Figure 4.1 shows how an detected

object which is partially occluded by the environment, shows various disparity measures. Top row represents frames the left camera images, and bottom row is disparity map extracted from the work of (Mayer et al., 2016). Even for low rates of occlusion mean disparity measure shows inconsistencies, where both the minimum and maximum disparity measure of the bounding box decreases the tracker performance. Further processing methods like occlusion-pose estimation (Wang et al., 2016), occlusion classifiers (Mathias et al., 2013) or histogram comparisons are required.

4.0.1 Bounding Box Geometry

For the set of extracted proposals D_T for frame T, bounding box features are extracted :

$$D_{T_i} = \{x_{mid}, y_{mid}, x_{width}, y_{height}\}, \forall i \in D_T \quad (4.1)$$

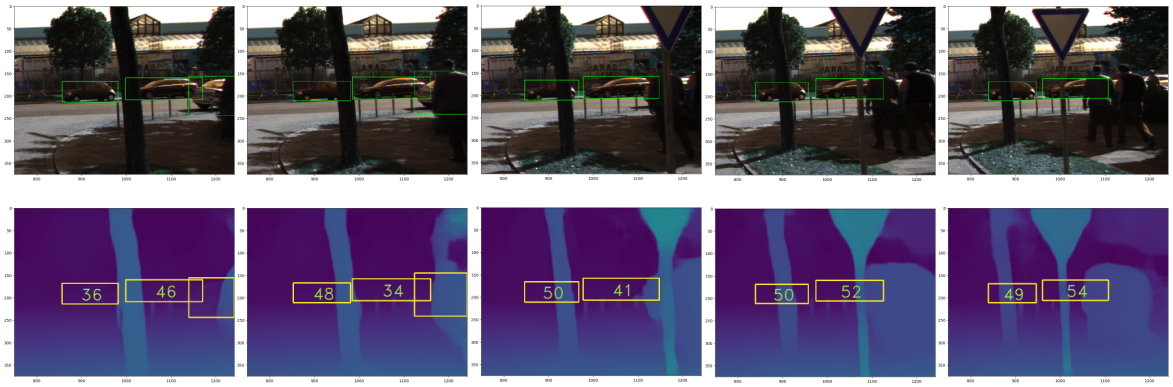

Figure 4.1: Top row shows detected vehicles and bottom row stereo disparity map for KITTI Object Tracking Training Sequence 0001, Frames 160 through 164. Mean disparity values of the area marked by the bounding box is shown in yellow. Even low rates of occlusion causes mean disparity measure of the detected object to fluctuate through the sequence, and does not provide a stable affinity feature.

Each feature is compared with the detections D_{T+1} of the next frame $T+1$ using Bray-Curtis Dissimilarity (BCD). For n variables with k as variable index, BCD is defined as :

$$BCD_{(i,j)} = \sum_{k=0}^{n-1} \frac{|n_{ik} - n_{jk}|}{(n_{ik} + n_{jk})} \quad (4.2)$$

Bounding box geometry dissimilarity matrix DS_{BB} for two consecutive frames is formed as :

$$DS_{BB_{i,j}} = BCD_{i,j}, \forall i \in D_T, \forall j \in D_{T+1} \quad (4.3)$$

4.0.2 Apperance Comparison

For affinity measure extraction based on appearance, 3-dimensional RGB histograms similarities are compared. Each channel is segmented into 5-bins with uniform ranges. The RGB histogram is normalized and flatten into one dimension, resulting an histogram of length $5 \times 5 \times 5$, that is normalized using l_2-norm. For the image patch I_i that is the defined by the bounding box of detection i, color histogram H_i is defined as :

$$
\begin{aligned}
H_i = &\big((I_{i_{R_{[0,1]},G_{[0,1]},B_{[0,1]})}}), (I_{i_{R_{[1,2]},G_{[0,1]},B_{[0,1]})}}), \ldots, \\
&(I_{i_{R_{[1,2]},G_{[1,2]},B_{[0,1]})}}), (I_{i_{R_{[1,2]},G_{[2,3]},B_{[0,1]})}}), \ldots, \\
&(I_{i_{R_{[3,4]},G_{[3,4]},B_{[2,3]})}}), (I_{i_{R_{[3,4]},G_{[3,4]},B_{[3,4]})}})\big)
\end{aligned}
\tag{4.4}
$$

Where $R_{[k,k+1]}$, $G_{[l,l+1]}$ and $B_{[m,m+1]}$ denotes the number of pixels founds in the corresponding channel bins. Similarity of color histograms of the object proposals between consecutive frames are compared using Chi-square test. For two color histograms H_i and H_j Chi-square test is defined as :

$$
\begin{aligned}
DS_{A_{i,j}} = &\chi^2(H_i, H_j) = \sum_I \frac{(H_i(I) - H_j(I))^2}{H_i(I)} \\
&\forall i \in D_T, \forall j \in D_{T+1}
\end{aligned}
\tag{4.5}
$$

Where DS_A is the final appearance based dissimilarity matrix.

4.0.3 Changing Scene

Characteristics that separates Multiple Object Tracking applications for autonomous driving from traditional Multiple Object Tracking applications is that the camera is not stationary. For dynamic scenes with high velocity objects and/or camera orientation changes using the changing scene information enhances data association by providing better affinity measurement between detected objects of consecutive frames. Changing scene is analyzed using the objects topology according to the shared keypoints from in the temporal window. For each frame ORB (Rublee et al., 2011) desciptors, which determine keypoints according to (Rosten and Drummond, 2006) and uses binary descriptors explained in (Calonder et al., 2010), are extracted. ORB descriptor extraction is faster relative to other methods like SIFT and SURF, with matching performance. ORB descriptors are suitable to our application due to the fact that a high number of desciptors are needed to ensure avaibility of shared ones in the temporal window.

For three consecutive frames in the temporal window and orb descriptors extracted from them kp_T, kp_{T+1} and kp_{T+2}, the set of matching keypoints $kp_\cap$ are extracted :

$$kp_{\cap} = (kp_T \cap kp_{T+1} \cap kp_{T+2}) \quad (4.6)$$

and for all detections i,j,k belonging to frames T, $T+1$ and $T+2$ respectively, distance of bounding box vertices to each of the keypoint in $kp_{\cap}$ are extracted :

$$d_{i,kp_m} = \{|x_{i_{min}} - x_{k_m}| + |y_{i_{min}} - y_{k_m}|, \quad (4.7)$$

$$|x_{i_{max}} - x_{k_m}| + |y_{i_{min}} - y_{k_m}|, \quad (4.8)$$

$$|x_{i_{min}} - x_{k_m}| + |y_{i_{max}} - y_{k_m}|, \quad (4.9)$$

$$|x_{i_{max}} - x_{k_m}| + |y_{i_{max}} - y_{k_m}|\}, kp_m \in kp_T \quad (4.10)$$

where d_i, d_j, d_k are vectors of length $4 \times ||k_{\cap}||$ representing the distance of bounding box vertices of objects i, j, k to the positions of all common keypoints in their respective frames T,$T+1$ and $T+2$:

$$d_i = \{d_{i,kp_\alpha} \ | \forall kp_\alpha \in kp_{\cap_T}\} \quad (4.11)$$

$$d_j = \{d_{j,kp_\beta} \ | \forall kp_\beta \in kp_{\cap_{T+1}}\} \quad (4.12)$$

$$d_k = \{d_{k,kp_\gamma} \ | \forall kp_\gamma \in kp_{\cap_{T+2}}\} \quad (4.13)$$

Detected objects dissimilarity matrix relating to the changing scene, DS_{CS}, for two consecutive frames and for detections $i \in T$, $j \in T+1$ is than :

$$DS_{CS_{i,j}} = \frac{\sum |d_i - d_j|}{||kp_{\cap}||} \quad (4.14)$$

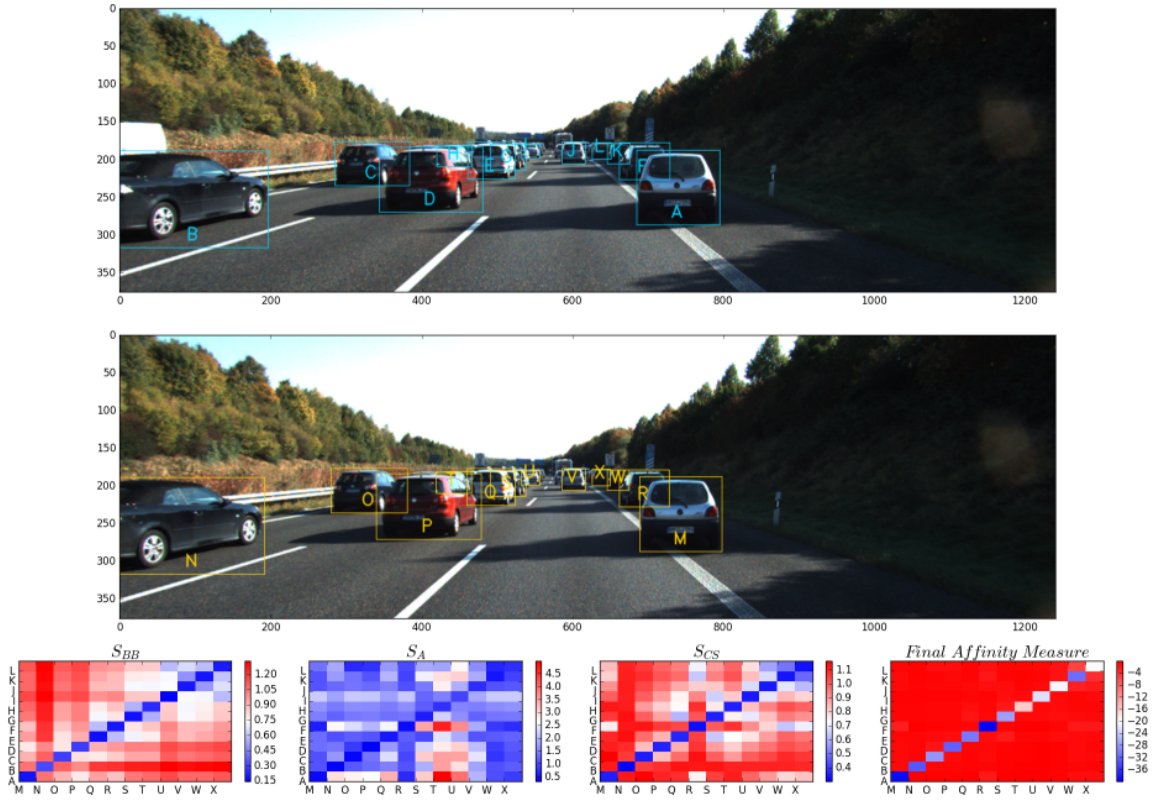

Figure 4.2: Affinity measures for KITTI training sequence 0020, frames 0000 and 0001.

First row shows the detected 'Car' class objects denoted 'A' to 'L' for the first frame. Second row are the next frames detections with objects 'M' to 'X'. Third row represents DS_{BB}, DS_A, DS_{CS} dissimilarity matrices and the final affinity cost matrix between objects.

4.1 Data Association

4.1.1 Bipartite matching

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 compated 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 4.3 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 matching of a bipartite graph introduced by the Hungarian Algorithm (Kuhn, 1955). 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 :

$$min \sum_{i} \sum_{j} C_{i,j} X_{i,j} \tag{4.15}$$

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.

4.1.2 Min-Cost Network Flow

Mostly min-cost flow formulations assume a batch setting, where the globally optimal solution is achieved. However it is also possible to use a sliding temporal window of fixed length, that would remove abiguities resulting from bipartite matching, i.e. data association between only two frames and perform online Multiple Object Tracking. In our work data association is performed by solving the min-cost flow problem of the temporal windows's affinity network. When dissimilarity matrices of detected objects are created for three consecutive frames $T-2$, $T-1$ and T an affinity network is

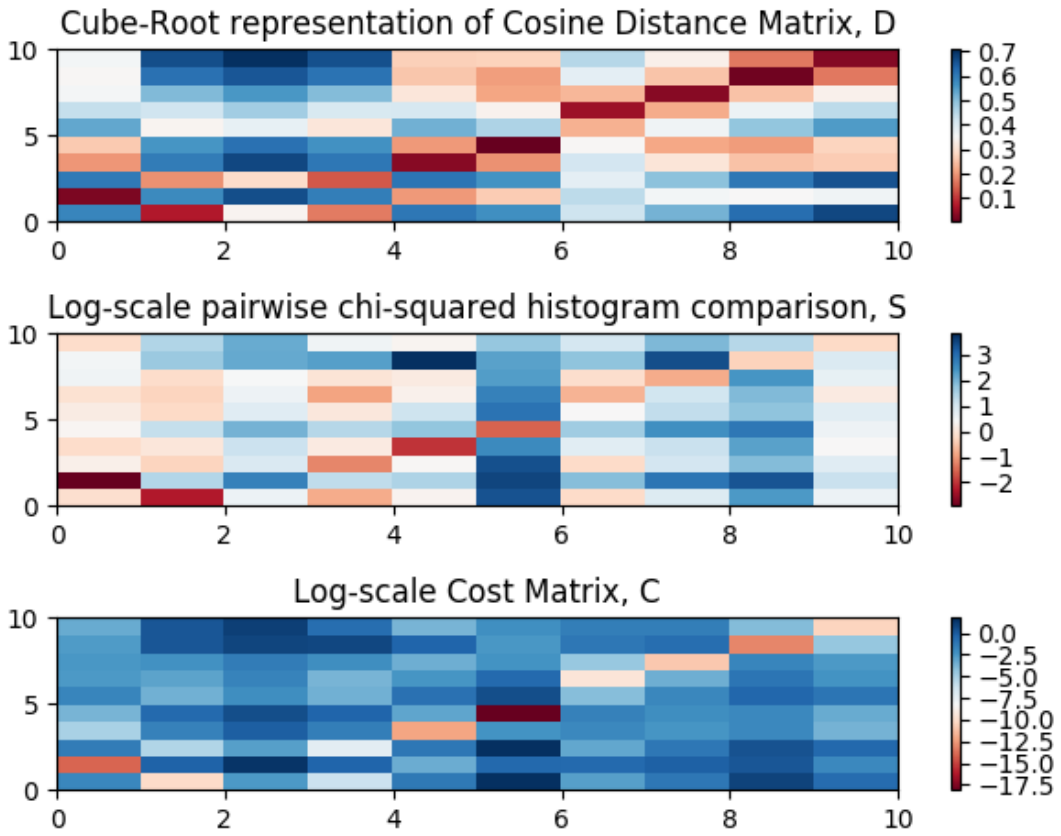

Figure 4.3: Feature cosine distance matrix, chi-squared RGB color histogram similarity and cost matrix is represented from top to bottom respectively. Row indexes represents detections of objects at frame t and columns indexes those of the previous frame, $t-1$.

initiated in order to associate the objects in T and decide on whether a tracked entitiy disappered. Decision is done whether an existing tracklet is preserved, terminated or whether a new tracklet is started. Network is composed of a directed graph with :

— "Tracklet Start" that is connected to all the object detection nodes with positive cost, initializing a tracklet, with positive cost.
— "Tracklet Terminate" that is connected to all the object detection nodes with positive cost, terminating a tracklet.
— "Object Detection Nodes", composed of two nodes with a single flow capacity
— "Affinity Edges", that connects object detection nodes of consecutive frames, with single flow capacity and negative cost.

For two detected objects i, j from two consecutives frames, an affinity edge $E_{i,j}$ with a negative cost $c_{i,j}$ is added to the graph if the dissimilarity $A_{i,j}$ is smaller than or equal to the edge threshold t_E :

$$A_{i,j} = DS_{CS_{i,j}} \times DS_{BB_{i,j}} \times DS_{A_{i,j}} \quad (4.16)$$

$$A_{i,j} \leq t_E \rightarrow E_{i,j} = c_{i,j} = \frac{-1}{A_{i,j}} \quad (4.17)$$

Also threshold t_N is set so that an object detection node has at most an out-degree of $t_N + 1$, limiting the number of edges to make the network sparser. Figure 4.4 shows an example affinity network for the detections of the temporal window. Green nodes represent object detection nodes that ensure a single flow is passed through any detected object, blue edges represents "affinity edges", while red node S "Tracklet Start" initializes tracklets and black node T terminates them.

Min-cost max flow problem on the established network structure decides on the detected objects at time T by either : assigning to an existing track if they are connected to any objects from $T-1$ or if not a new track is started. Any existing track that is not connected to objects of T are defined inactive and cached.

4.1.3 Cached Vehicles

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

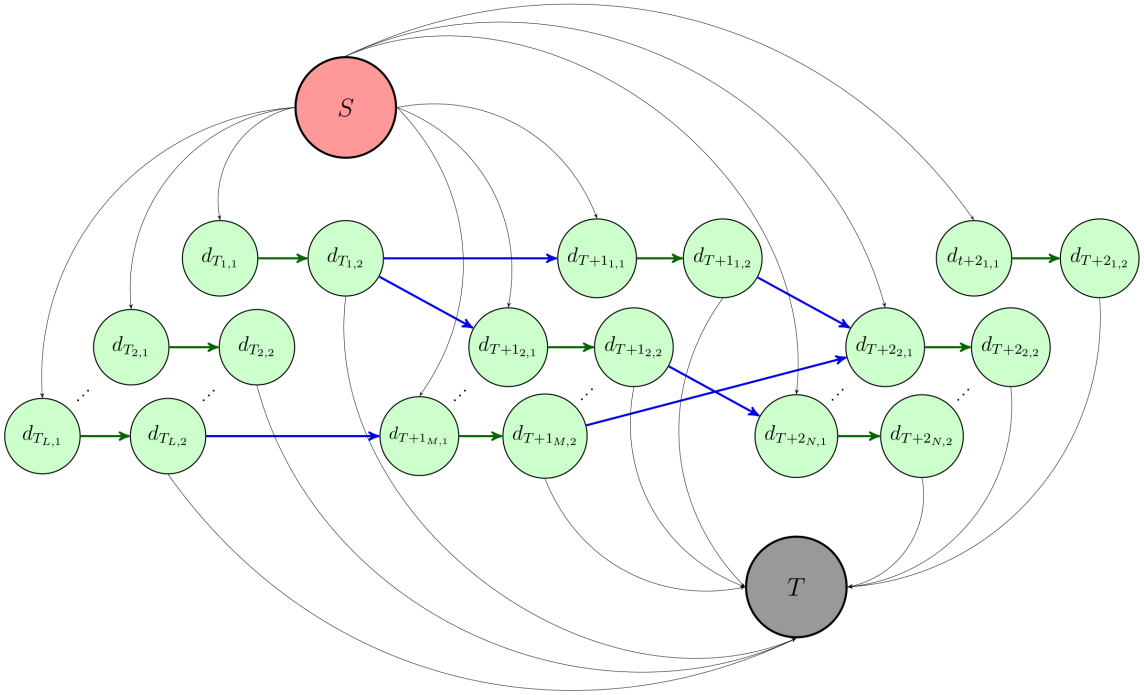

Figure 4.4: Network diagram, where green nodes represents object detection nodes, S Tracklet Start and T Tracklet Terminate node. Blue edges are "affinity edges" with negative costs that are between detected objects of two consecutive frames.

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 4.6, 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.

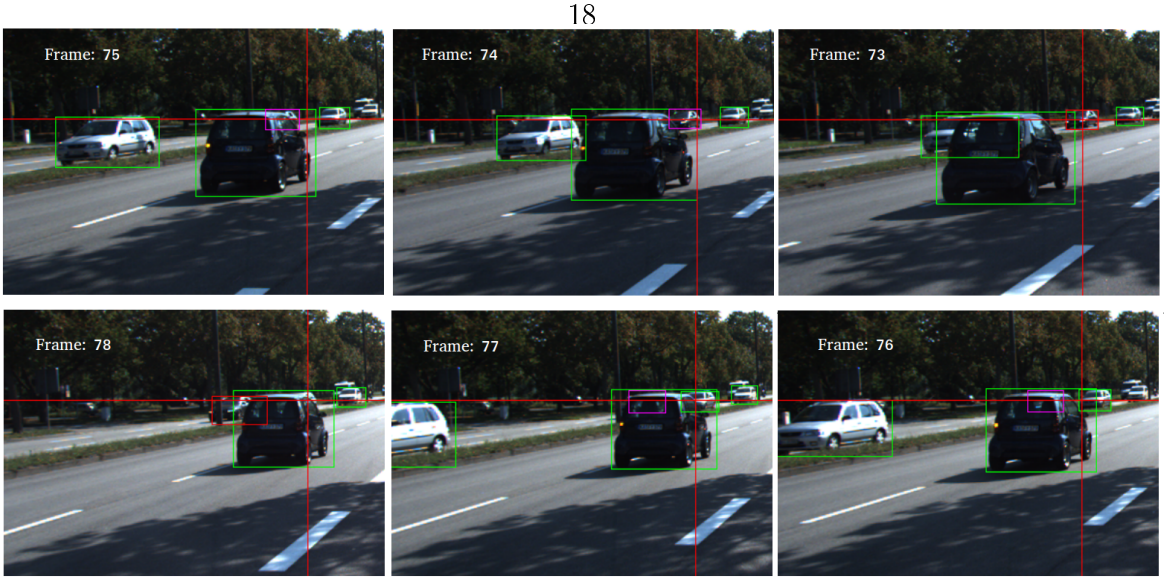

Figure 4.5: KITTI Object Tracking Testing Sequence 0007, Frames 73 through 78. Top right image shows the last detection of a vehicle, where red horizontal and vertical lines represents the bounding box coordinates. For frames 74-77 purple bounding box is the predicted movement of the same vehicle. Left most figure, Frame 78 shows the bounding box when the same vehicle is detected.

The frame placed in the bottom left of Figure 4.6 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 convolutional neural network. An illustrative example is shown at Figure 4.5.

Same vehicle is last detected by the convolutional neural network at frame 73, which is plotted in the rightmost 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.

Objects with terminated tracks are cached for a number of frames. For each frame they are not detected, the line of best fit, least squares method, is generated for the last 5 bounding box features known. And for each cached object current frame bounding box features are extrapolated. Before a new track is started the detected object that has not been associated with the detections of $T-1$, is compared to the cached ones with just respect to DS_{BB} and if the dissimilarity is smaller or equal to d_C than the object is assigned to the cached track.

Table 4.1: List of optimized parameters.

Parameter	Range	
t_{d_1}	[0.06,0.15]	Threshold for object detection.
t_{d_2}	[0.1,0.9]	Threshold for acceptance of connected components with single proposal.
t_{IoU}	[0.4,0.9]	Threshold for minimum intersection over union value to create connected component.
t_E	[0.001,0.8]	Maximum dissimilarity for a possible edge threshold.
d_C	[0.09, 0.2]	Maximum acceptance distance with cached tracklets.
t_N	[1,3]	Maximum number of edges an object detection node can have.

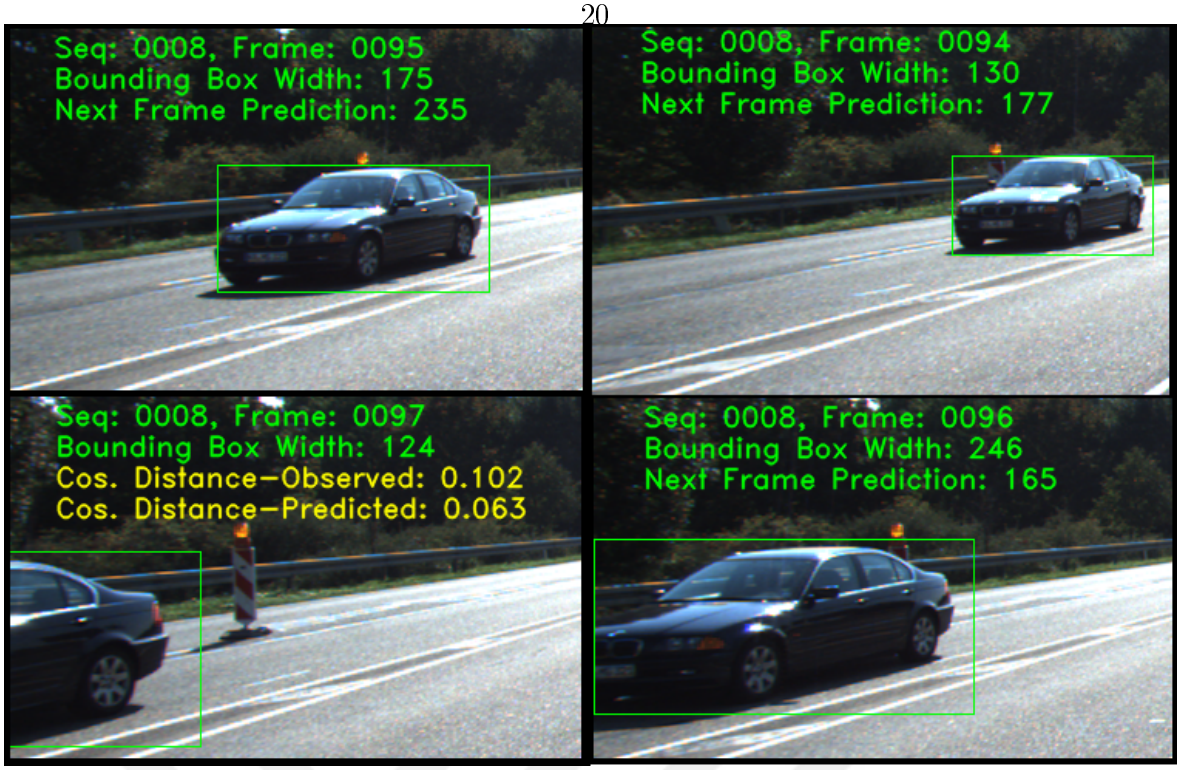

Figure 4.6: The sequence of frames numbered 94 through 97 belonging to Tracking Training Sequence 0008 represents considerable bounding box size changes between consecutive frames. For affinity matching, bounding box of a tracked object that is expected to be partially visible the following frame is adjusted.

4.2 Experimental Evaluation

Proposed Multiple Object Tracking method is evaluated on KITTI Object Tracking Evaluation 2012 dataset for the 'Car' class. KITTI Object Tracking dataset consists of 21 training sequences with 8.008 frames and 29 testing sequences with 11.095 frames. Frames were recorded at 10 FPS from a camera mounted on the 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 : Recall, Precision, F1-Measure, False Alarm Rate, False Positives, False Negatives relating to the object detection task and Runtime, widely used CLEAR MOT (Bernardin and Stiefelhagen, 2008) metrics like Multiple Object Tracking Accuracy (MOTA) and also Fragmentation (FRAG),ID-switch (IDS), Mostly-Tracked (MT) and Mostly-Lost (ML) defined in (Li et al., 2009) for Multiple Object Tracking evaluation.

To give a brief summary of the metrics, MOTA is the ratio of the total sum of FN, FP and mismatches computed over the benchmark versus the total number of ground truth objects, Multiple Object Tracking Precision (MOTP) illustrates the precise object

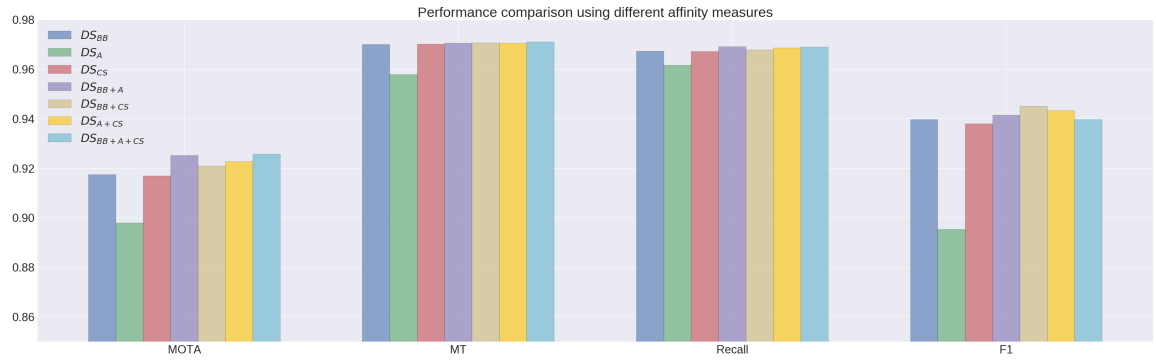

Figure 4.7: MOTA, MT, Recall and F1 metrics evaluated on training dataset for affinity measure subsets of DS_{BB}, DS_A and DS_{CS}. For MOTA metric using just bounding box affinity performs the best. It is enhanced when combined with relating to the changing scene while best MOTA is achieved when all three features are used on data association.

position estimation ability of the tracke. 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 GT trajectory changed its identity and FRAG defines the number of times a ground truth trajectory is interrupted.

Differential Evolution (DE) (Price et al., 2006) has been used to optimize and learn the previously explained parameters on the training dataset. Table 4.1 shows defined minimum and maximum ranges for the parameters. A population size of 50 were created using latin-hypercube sampling (Stein, 1987) and using DE notation *best/2/bin*, further information is availible at (Price et al., 2006). In Figure 4.7 how the subset of three affinity measure performed on training set relating to the MOTA, MT, Recall and F1 metrics. When used individually DS_{BB} reported the highest MOTA and MT values followed by DS_{CS} while DS_A performed worse. When DS_{BB} and DS_{CS} are used together, denoted by DS_{BB+CS}, they report a higher MOTA value by 0.925218, but the best result is achieved when all three afinity measures are used together that reported a MOTA value of 0.925675.

Table 4.5 shows state-of-the-art tracker evaluation results for "Car" class of KITTI Tracking Benchmark. Results are sorted for the MOTA metric, online methods are shown in bold and our method SASN-MCF is highlighted in blue ; Remaining methods adopt batch processing approaches. Our method ranks second on MOTA, MT, ML metrics within online methods with 83.10 %, 70.92 % and 3.85% respectively. While

Table 4.2: Mean runtimes and standard deviations of tracker components.

	IoUC	KPE	KPM	MCF	EC	Total
Mean	0.0028 s	0.0113 s	0.0045 s	0.0068 s	0.0068 s	0.0474 s
Std.	0.0019 s	0.0018 s	0.0005 s	0.0034 s	0.0119 s	0.0145 s

reporting lower IDS of 213 and FRAG of 702 values compared to the best MOTA performing method. While having a mean running time that is 6 times faster, at 20Hz.

Mean runtime and standart deviations of each tracker component for continious processing of the KITTI tracking sequences can be seen in Table 4.2, which are reported on an Intel i7-6820HK @ 2.7GHz CPU. IoUC represents ensemble bounding box selection process, KPE key point extraction, KPM key point match, MCF solving min-cost flow, EC feature extrapolation of cached tracklets and Total is total process including all the components and remaining tasks like frame retrival. Even though a brute-force keypoint matching is performed KPM is relatively fast, while most computation is required for the extracting of the high number of keypoint objects. Runtime of IoUC and MCF are also fast due to the sparse nature of the created network, with stable deviations. However EC component shows high variation for scenes with large number of inactive tracks.

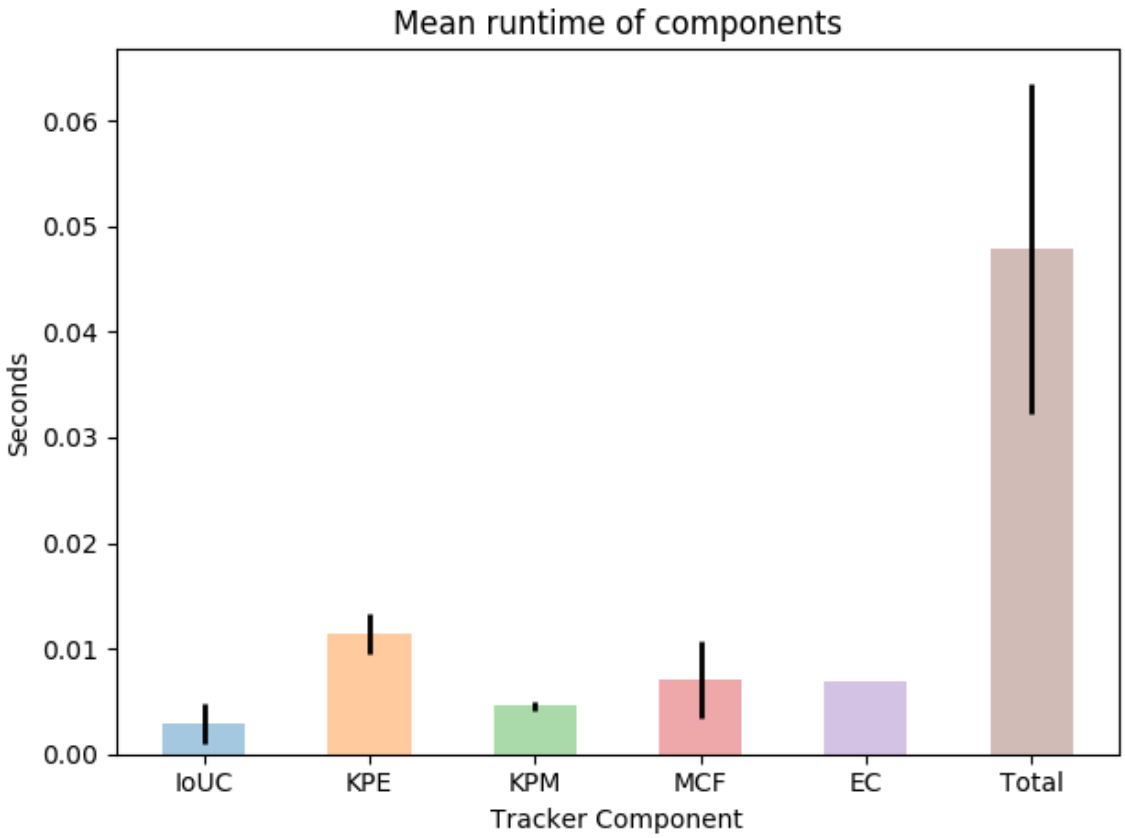

Figure 4.8: Runtime of tracker components. IoUC represents ensemble bounding box selection process, KPE key point extraction, KPM key point match, MCF solving min-cost flow, EC feature extrapolation of cached tracklets and Total is total process including all the components and remaining tasks like frame retrival.

Method	Setting	MOTA	MOTP	MT	ML	IDS	FRAG
iDST-VT		88.93 %	84.30 %	76.31 %	4.77 %	134	289
youtu		88.48 %	84.85 %	76.77 %	3.54 %	181	550
TuSimple		86.62 %	83.97 %	72.46 %	6.77 %	293	501
NECMA		84.98 %	83.14 %	70.77 %	9.08 %	33	162
RRC-IIITH	on	84.24 %	85.73 %	73.23 %	2.77 %	468	944
SASN-MCF	on	83.10 %	81.87 %	70.92 %	3.85 %	213	702
IMMDP	on	83.04 %	82.74 %	60.62 %	11.38 %	172	365
DF-PC_CNN	la on	80.74 %	84.72 %	61.08 %	6.31 %	155	983
JCSTD	on	80.57 %	81.81 %	56.77 %	7.38 %	61	643
3D-CNN/PMBM	gp on	80.39 %	81.26 %	62.77 %	6.15 %	121	613
extraCK	on	79.99 %	82.46 %	62.15 %	5.54 %	343	938
MCMOT-CPD		78.90 %	82.13 %	52.31 %	11.69 %	228	536
NOMT*		78.15 %	79.46 %	57.23 %	13.23 %	31	207
LP-SSVM*		77.63 %	77.80 %	56.31 %	8.46 %	62	539
MDP	on	76.59 %	82.10 %	52.15 %	13.38 %	130	387
DSM		76.15 %	83.42 %	60.00 %	8.31 %	296	868
SCEA*	on	75.58 %	79.39 %	53.08 %	11.54 %	104	448
CIWT*	st on	75.39 %	79.25 %	49.85 %	10.31 %	165	660
NOMT-HM*	on	75.20 %	80.02 %	50.00 %	13.54 %	105	351
SSP*		72.72 %	78.55 %	53.85 %	8.00 %	185	932
mbodSSP*	on	72.69 %	78.75 %	48.77 %	8.77 %	114	858
MTCCF		71.27 %	81.38 %	48.31 %	5.85 %	537	1018
TENSOR		71.18 %	79.15 %	47.85 %	11.69 %	418	947
MBKF	on	69.77 %	83.03 %	41.23 %	11.38 %	410	971

Table 4.3: 'Car' class MOTA, MOTP, MT, ML, IDS and FRAG metrics for submitted methods on KITTI Object Tracking Evaluation 2012 benchmark.

Method	Setting	MOTA	MOTP	MT	ML	IDS	FRAG
MBKF	on	69.77 %	83.03 %	41.23 %	11.38 %	410	971
DCO-X*		68.11 %	78.85 %	37.54 %	14.15 %	318	959
NOMT		66.60 %	78.17 %	41.08 %	25.23 %	13	150
RMOT*	on	65.83 %	75.42 %	40.15 %	9.69 %	209	727
LP-SSVM		61.77 %	76.93 %	35.54 %	21.69 %	16	422
NOMT-HM	on	61.17 %	78.65 %	33.85 %	28.00 %	28	241
ODAMOT	on	59.23 %	75.45 %	27.08 %	15.54 %	389	1274
SSP		57.85 %	77.64 %	29.38 %	24.31 %	7	704
SCEA	on	57.03 %	78.84 %	26.92 %	26.62 %	17	461
mbodSSP	on	56.03 %	77.52 %	23.23 %	27.23 %	0	699
TBD		55.07 %	78.35 %	20.46 %	32.62 %	31	529
RMOT	on	52.42 %	75.18 %	21.69 %	31.85 %	50	376
CEM		51.94 %	77.11 %	20.00 %	31.54 %	125	396
MCF		45.92 %	78.25 %	14.92 %	37.23 %	21	581
HM	on	43.85 %	78.34 %	12.46 %	39.54 %	12	571
DP-MCF		38.33 %	78.41 %	18.00 %	36.15 %	2716	3225
DCO		37.28 %	74.36 %	15.54 %	30.92 %	220	612
FMMOVT		31.88 %	77.68 %	21.38 %	34.92 %	511	930

Table 4.4: Remaining 'Car' class MOTA, MOTP, MT, ML, IDS and FRAG metrics for submitted methods on KITTI Object Tracking Evaluation 2012 benchmark.

Method	MOTA	MOTP	MT	ML	IDS	FRAG	Runtime	Recall	Precision	F1	TP	FP	FN	FAR
iDST-VT	88.93 %	84.30 %	76.31 %	4.77 %	134	289	0.5 s	92.08 %	98.36 %	95.12 %	35775	596	3078	5.36
youtu	88.48 %	84.85 %	76.77 %	3.54 %	181	550	0.6 s	91.69 %	98.36 %	94.91 %	35219	587	3193	5.28 %
TuSimple (Choi, 2015)	86.62 %	83.97 %	72.46 %	6.77 %	293	501	0.6 s	90.50 %	97.99 %	94.10 %	34322	705	3602	6.34 %
NECMA	84.98 %	83.14 %	70.77 %	9.08 %	33	162	0.5 s	88.70 %	97.90 %	93.07 %	34487	738	4395	6.63 %
RRC-IIITH (Sharma et al., 2018)	**84.24 %**	**85.73 %**	**73.23 %**	**2.77 %**	**468**	**944**	**0.3 s**	**88.80 %**	**97.95 %**	**93.15 %**	**33656**	**705**	**4247**	**6.34 %**
SASN-MCF	**83.10 %**	**81.87 %**	**70.92 %**	**3.85 %**	**213**	**702**	**0.05 s**	**88.25 %**	**96.84 %**	**92.35 %**	**33778**	**1101**	**4499**	**9.90 %**
IMMDP (Xiang et al., 2015)	**83.04 %**	**82.74 %**	**60.62 %**	**11.38 %**	**172**	**365**	**0.19 s**	**86.11 %**	**98.82 %**	**92.03 %**	**32668**	**391**	**5269**	**3.51 %**
DF-PC_CNN	**80.74 %**	**84.72 %**	**61.08 %**	**6.31 %**	**155**	**983**	**0.01 s**	**85.38 %**	**97.30 %**	**90.95 %**	**32516**	**904**	**5566**	**8.13 %**
JCSTD	**80.57 %**	**81.81 %**	**56.77 %**	**7.38 %**	**61**	**643**	**0.11 s**	**83.37 %**	**98.72 %**	**90.40 %**	**31162**	**405**	**6217**	**3.64 %**
3D-CNN/PMBM	**80.39 %**	**81.26 %**	**62.77 %**	**6.15 %**	**121**	**613**	**0.01 s**	**85.01 %**	**96.93 %**	**90.58 %**	**31841**	**1007**	**5616**	**9.05 %**
extraCK	**79.99 %**	**82.46 %**	**62.15 %**	**5.54 %**	**343**	**938**	**0.03 s**	**84.51 %**	**98.04 %**	**90.77 %**	**32156**	**642**	**5896**	**5.77 %**
MCMOT-CPD (Lee et al., 2016)	78.90 %	82.13 %	52.31 %	11.69 %	228	536	0.01 s	81.84 %	98.97 %	89.59 %	30247	316	6713	2.84 %
NOMT* (Choi, 2015)	78.15 %	79.46 %	57.23 %	13.23 %	31	207	0.09 s	83.22 %	96.78 %	89.49 %	31854	1061	6421	9.54 %
LP-SSVM* (Wang and Fowlkes, 2017)	77.63 %	77.80 %	56.31 %	8.46 %	62	539	0.02 s	83.35 %	96.27 %	89.34 %	31997	1239	6393	11.14 %
MDP (Xiang et al., 2015)	**76.59 %**	**82.10 %**	**52.15 %**	**13.38 %**	**130**	**387**	**0.9 s**	**80.26 %**	**98.00 %**	**88.25 %**	**29747**	**606**	**7315**	**5.45 %**
DSM	76.15 %	83.42 %	60.00 %	8.31 %	296	868	0.1 s	80.23 %	98.09 %	88.27 %	29736	578	7328	5.20 %
SCEA* (Hong Yoon et al., 2016)	**75.58 %**	**79.39 %**	**53.08 %**	**11.54 %**	**104**	**448**	**0.06 s**	**81.76 %**	**96.00 %**	**88.31 %**	**31330**	**1306**	**6989**	**11.74 %**
CIWT* (Osep et al., 2017)	**75.39 %**	**79.25 %**	**49.85 %**	**10.31 %**	**165**	**660**	**0.28 s**	**80.32 %**	**96.92 %**	**87.84 %**	**29985**	**954**	**7345**	**8.58 %**
NOMT-HM* (Choi, 2015)	**75.20 %**	**80.02 %**	**50.00 %**	**13.54 %**	**105**	**351**	**0.09 s**	**83.22 %**	**96.78 %**	**89.49 %**	**31854**	**1061**	**6421**	**9.54 %**
SSP* (Lenz et al., 2015)	72.72 %	78.55 %	53.85 %	8.00 %	185	932	0.6 s	82.69 %	92.57 %	87.35 %	31764	2548	6648	22.91 %

Table 4.5: Metric comparison for state-of-the-art Multiple Object Tracking applications, online methods are shown in bold, while our method SASN-MCF is highlighted in blue.

5 CONCLUSION

In this work we have proposed an Multiple Object Tracking application that is specified for the needs of autonomous driving : computational efficiency, being able to perform Multiple Object Tracking on data acquired from non-stationary camera and most importantly track object from a stream of data, i.e. being an online method. Efficieny was established using strong affinity between the detected objects from the temporal window relating to a number of dissimilarity metrics that each has its purpose for relating scenarios ; where the strong affinity made possible to create a sparse network that min-cost max flow problem could be efficienty solved.

The proposed method was validated on KITTI Tracking 'Car' class benchmark and reported state-of-the-art performance on online methods category. While having a mean running time of 0.0474 s and a standard deviation of 0.0145 that could assist tasks like motion planning of autonomous driving applications, while performing second on MOTA, MT, ML metrics and best of IDS and FRAG on online methods.

REFERENCES

Bernardin, K. and Stiefelhagen, R. (2008). Evaluating multiple object tracking performance : the clear mot metrics, *EURASIP Journal on Image and Video Processing* **2008**(1) : 246309.

Calonder, M., Lepetit, V., Strecha, C. and Fua, P. (2010). Brief : Binary robust independent elementary features, *European conference on computer vision*, Springer, pp. 778–792.

Chen, L., Ai, H., Shang, C., Zhuang, Z. and Bai, B. (2017). Online multi-object tracking with convolutional neural networks, *Image Processing (ICIP), 2017 IEEE International Conference on*, IEEE, pp. 645–649.

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

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

Darms, M., Rybski, P. and Urmson, C. (2008). Classification and tracking of dynamic objects with multiple sensors for autonomous driving in urban environments, *Intelligent Vehicles Symposium, 2008 IEEE*, IEEE, pp. 1197–1202.

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

Girshick, R., Donahue, J., Darrell, T. and Malik, J. (2014). Rich feature hierarchies for accurate object detection and semantic segmentation, *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 580–587.

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

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

Hosang, J., Benenson, R. and Schiele, B. (2017). Learning non-maximum suppression, *arXiv preprint* .

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

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

Krizhevsky, A., Sutskever, I. and Hinton, G. E. (2012). Imagenet classification with deep convolutional neural networks, *Advances in neural information processing systems*, pp. 1097–1105.

Kuhn, H. W. (1955). The hungarian method for the assignment problem, *Naval Research Logistics (NRL)* **2**(1-2) : 83–97.

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

Lenz, P., Geiger, A. and Urtasun, R. (2015). Followme : Efficient online min-cost flow tracking with bounded memory and computation, *Proceedings of the IEEE International Conference on Computer Vision*, pp. 4364–4372.

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

Li, Y., Huang, C. and Nevatia, R. (2009). Learning to associate : Hybridboosted multi-target tracker for crowded scene, *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*, IEEE, pp. 2953–2960.

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

Mayer, N., Ilg, E., Häusser, P., Fischer, P., Cremers, D., Dosovitskiy, A. and Brox, T. (2016). A large dataset to train convolutional networks for disparity, optical flow, and scene flow estimation, *IEEE International Conference on Computer Vision and Pattern Recognition (CVPR)*. arXiv :1512.02134.
URL: *http ://lmb.informatik.uni-freiburg.de/Publications/2016/MIFDB16*

Osep, A., Mehner, W., Mathias, M. and Leibe, B. (2017). Combined image-and world-space tracking in traffic scenes, *Robotics and Automation (ICRA), 2017 IEEE International Conference on*, IEEE, pp. 1988–1995.

Petrovskaya, A. and Thrun, S. (2009). Model based vehicle detection and tracking for autonomous urban driving, *Autonomous Robots* **26**(2-3) : 123–139.

Price, K., Storn, R. M. and Lampinen, J. A. (2006). *Differential evolution : a practical approach to global optimization*, Springer Science & Business Media.

Redmon, J., Divvala, S., Girshick, R. and Farhadi, A. (2016). You only look once : Unified, real-time object detection, *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 779–788.

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

Rosten, E. and Drummond, T. (2006). Machine learning for high-speed corner detection, *European conference on computer vision*, Springer, pp. 430–443.

Rothe, R., Guillaumin, M. and Van Gool, L. (2014). Non-maximum suppression for object detection by passing messages between windows, *Asian Conference on Computer Vision*, Springer, pp. 290–306.

Rousseeuw, P. J. (1987). Silhouettes : a graphical aid to the interpretation and validation of cluster analysis, *Journal of computational and applied mathematics* **20** : 53–65.

Rublee, E., Rabaud, V., Konolige, K. and Bradski, G. (2011). Orb : An efficient alternative to sift or surf, *Computer Vision (ICCV), 2011 IEEE international conference on*, IEEE, pp. 2564–2571.

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

Sharma, S., Ansari, J. A., Murthy, J. K. and Krishna, K. M. (2018). Beyond pixels : Leveraging geometry and shape cues for online multi-object tracking, *arXiv preprint arXiv :1802.09298* .

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

Stein, M. (1987). Large sample properties of simulations using latin hypercube sampling, *Technometrics* **29**(2) : 143–151.

Sun, H., Sun, X., Wang, H., Li, Y. and Li, X. (2012). Automatic target detection in high-resolution remote sensing images using spatial sparse coding bag-of-words model, *IEEE Geoscience and Remote Sensing Letters* **9**(1) : 109–113.

Szegedy, C., Reed, S., Erhan, D., Anguelov, D. and Ioffe, S. (2014). Scalable, high-quality object detection, *arXiv preprint arXiv :1412.1441* .

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

Tibshirani, R., Walther, G. and Hastie, T. (2001). Estimating the number of clusters in a data set via the gap statistic, *Journal of the Royal Statistical Society : Series B (Statistical Methodology)* **63**(2) : 411–423.

Urmson, C., Anhalt, J., Bagnell, D., Baker, C., Bittner, R., Clark, M., Dolan, J., Duggins, D., Galatali, T., Geyer, C. et al. (2008). Autonomous driving in urban environments : Boss and the urban challenge, *Journal of Field Robotics* **25**(8) : 425–466.

Wang, B., Wang, G., Chan, K. L. and Wang, L. (2017). Tracklet association by online target-specific metric learning and coherent dynamics estimation, *IEEE transactions on pattern analysis and machine intelligence* **39**(3) : 589–602.

Wang, C., Fang, Y., Zhao, H., Guo, C., Mita, S. and Zha, H. (2016). Probabilistic inference for occluded and multiview on-road vehicle detection, *IEEE Transactions on Intelligent Transportation Systems* **17**(1) : 215–229.

Wang, S. and Fowlkes, C. C. (2017). Learning optimal parameters for multi-target tracking with contextual interactions, *International Journal of Computer Vision* **122**(3) : 484–501.

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

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

BIOGRAPHICAL SKETCH

Gultekin Gunduz

— Izmir American College 2005-2009

— Computer Science and Engineering, Sabanci University 2009-2014

— Computer Engineering, Galatasaray University 2015-2018