IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received July 8, 2019, accepted July 26, 2019, date of publication July 31, 2019, date of current version August 14, 2019.

Digital Object Identifier 10.1109/ACCESS.2019.2932301

Multiple Object Tracking With Attention to Appearance, Structure, Motion and Size

HASITH KARUNASEKERA, HAN WANG, (Senior Member, IEEE), AND HANDUO ZHANG
School of Electrical and Electronic Engineering, Nanyang Technological University, Singapore 639798
Corresponding author: Han Wang (hw@ntu.edu.sg)

ABSTRACT Objective of multiple object tracking (MOT) is to assign a unique track identity for all the objects of interest in a video, across the whole sequence. Tracking-by-detection is the most common approach used in addressing MOT problem. In this work, we propose a method to address MOT by defining a dissimilarity measure based on object motion, appearance, structure, and size. We calculate the appearance and structure-based dissimilarity measure by matching histograms following a grid architecture. Motion and size for each track are predicted using the information from track's history. These dissimilarity values are then used in the Hungarian algorithm, in the data association step for track identity assignment. In addition, we introduce a method to address any false detection in stable tracks. The proposed method runs in real time following an online approach. We evaluate our method in both MOT17 benchmark data-set for pedestrian tracking and KITTI benchmark data-set for vehicle tracking using the same system parameters to verify the robustness of the proposed method. The method can achieve state-of-the-art results in both benchmarks.

INDEX TERMS Multiple object tracking, grid-based histograms, tracking by detection, online tracking, multiple car tracking, multiple human tracking.

I. INTRODUCTION

Tracking is a challenging problem in many video analysis tasks where an object (defined by a bounding box), is to be identified and assigned a unique identity over all the frames it appears in an image sequence. Tracking is an important task in many applications such as surveillance, autonomous driving & advance driver assistance systems, behavior analysis, motion prediction and particle transformation analysis.

Tracking research can broadly be divided in to multiple Object Tracking (MOT) and single object tracking. While MOT assumes object detection as prior knowledge, the latter tries to localize and track an unknown object that has only been described by the localization information at the first frame. In the recent years, the most prominent technique in single object tracking follow *discriminative method*, opposed to generative methods. Instead of building an object appearance model based on generative process and without considering the background [1], discriminative trackers are able to distinguish the target from negative samples by learning a classifier, which is more accurate [2]. TLD tracker [3] divides tracking process into three stages (tracking, learning and detecting); Correlation filter (CF) based tracking [4] and kernelized correlation filter (KCF) based tracking [5] (by extending the linear filter into non-linear space) can be presented as significant achievements. With the recent break through in deep learning domain, deeply learned single object trackers [6], [7] are developed.

On the other hand MOT tries to track multiple objects present in all the frames over a given image sequence. MOT generally has an object detector to detect the objects in each frame and then utilizes a detection association method to track them over time. Almost all the MOT methods follows such tracking-by-detection framework. Methods used in MOT can be separated to *Online methods* and *Offline methods* according to how they use object detection information in the image sequence. Offline methods make use of all detections available from the whole image sequence and handle tracking as a global optimization problem when associating unique track identities to these detections. Therefore, offline methods [8], [9] can only be applied to situations where the whole image sequence is present. In contrast, online methods only rely on the information from object detection up to the current frame, which makes it suitable for real time applications. Offline methods have additional information on the objects in the whole sequence and hence generally show a

The associate editor coordinating the review of this manuscript and approving it for publication was Tao Zhou.

higher accuracy compared to online methods. However, only online methods can be used in real time applications and have shown competitive tracking accuracy in both [10] and KITTI [11] benchmarks.

In this paper we propose an online tracker which can be used for multiple class object tracking. In the proposed method we use a detector to detect the object bounding boxes in an image frame and associate these detections to tracks using Hungarian Algorithm [12] based on pair-wise dissimilarity scores calculated between detections in the current frame against the tracks in the memory. The dissimilarity cost is calculated using four distance measures, considering appearance, structure, motion and size for each track. Correlation between two color histograms of the track and the detection is used as the appearance based similarity measure. Dissimilarity measure on structure is calculated from Linear Binary Pattern Histogram (LBPH) matching between the track and the current detection. Motion dissimilarity is based on the distance between predicted object location and observed detection location. The last dissimilarity component is calculated based on the Intersection over Union (IoU) between the predicted bounding box size and the current detection bounding box size. Additionally, we introduce a method to address false negatives in any stable tracks due to failures in detection. This overall tracking process is presented as a flow chart in Fig. 1.

In the next section we present the literature review specifically on online MOT domain, followed by the details on the proposed tracking methodology. Then in the fourth section we evaluate our method in terms of multiple pedestrian and car tracking in public benchmarks and discuss the results, followed by concluding remarks.

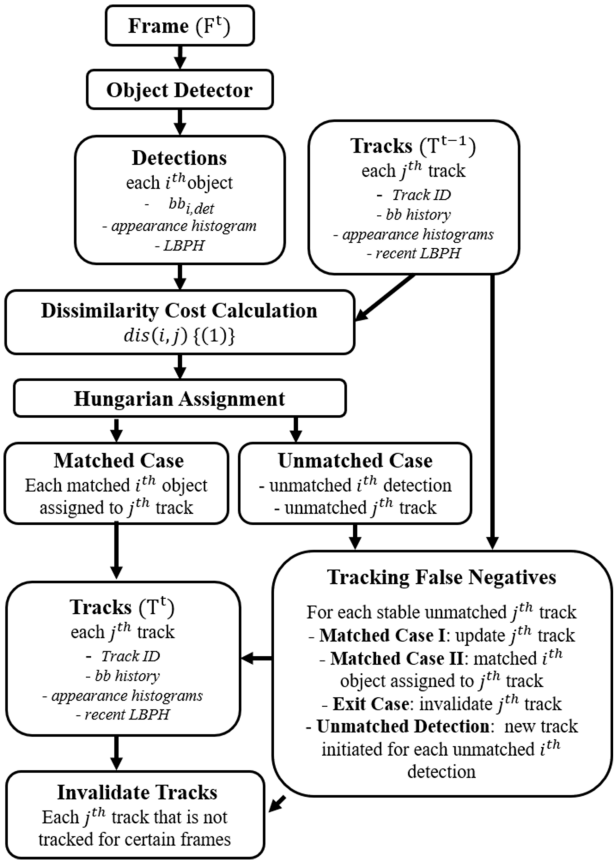

FIGURE 1. Flow chart of the proposed tracking framework, where $bb_{i,det}$ is the bounding box of the i^{th} detection, *LBPH* stands for Linear Binary Pattern Histogram and $dis(i,j)$ is the dissimilarity cost between i^{th} object detected in the current frame with the j^{th} object in the track list as defined in (1).

II. RELATED WORK

MOT is a widely studied area in the recent past. While some MOT methods are designed for specific object classes, such as multiple human tracking [9], [13], [14], some methods [8], [15]–[17] are class agnostic. Most MOT methods follows tracking-by-detection framework where they rely on an object detector to provide object candidates. Some methods that follow tracking-by-detection framework make use of all the object candidates in the sequence while the others use candidates up to that frame. Former methods are classified as global while the later are online methods. Network flow optimization [9], graph based clustering [13], multiple hypotheses tracking (MHT) [18] and Bayesian filtering based tracking [8] are amongst the popular approaches in global MOT methods in the recent past. In addition to the direct association of detections to tracks, some global MOT methods [8] first assign detections to tracklets (which are a combination of matching detections in few consecutive frames) and then assign tracklets to tracks to address long term variations in track objects. When calculating the matching cost or dissimilarity between detections in different frames, object's appearance and motion are the most common information sources. In appearance based distance calculations, traditional methods such as distance between RGB color histograms [8], distance between deeply learned features [9], [13], [18] and deeply learned person re-identification [9] are used. Point motion matching [8], distance between expected and detected positions [13], [18] and spatio-temporal distance between object bounding boxes [9] are used to define motion based matching costs. In addition, as a post processing step change point detection framework is used [8] rectify errors in tracklets to track assignment. Some methods use [13] multiple detectors to improve the tracking results, at the expense of increased computational time.

While offline MOT methods mostly use some form of global optimization to associate objects or tracklets to tracks using matching costs from all the frames, online MOT methods operate on information up to current frame. Therefore, online methods are more suitable for real time applications. Online MOT methods mostly use pair-wise cost to match the detections to tracks. Similar to global methods, matching cost calculation is based on appearance and motion information from the detections. Since the proposed framework is an online pair-wise cost calculation based method, few of such methods are discussed in detail. Sarthak Sharma

et al. in [16] use object shape, pose, 2D and 3D localization information and deeply learned keypoints matching followed by Hungarian assignment [12] in its tracking framework. Two object detectors [19], [20] are used to improve detection accuracy. In [21] dissimilarity measure is calculated using cosine distance of the spatio-temporal location and Chi-Square similarity of the RGB histogram of the object. Motion cues in matching cost calculations, can give rise to false matching when the camera is subjected to a sudden motion. This global motion can be compensated by using relative structural motion cues between object bounding boxes [17]. As a post processing step, [17] introduces an event aggregation to cater for false positives and false negatives. Reinforcement learning based pair wise similarity calculation is used in [22]. Amir *et al.* [14] presented an online method that uses three recurrent neural networks (RNNs) to calculate appearance, motion and social interaction model based similarity scores. In [23] authors use deeply learned person re-identification scores and object localization information in a Recurrent Autoregressive Network to calculate pair-wise costs. Chen *et al.* [24] propose to use the bounding boxes from both object detector and predicted bounding box (from each track's history) to reduce false negatives and define the pair-wise cost based on deeply learned person re-identification score. Detections are filtered using non maximum suppression using a cost define based on detection confidence and track confidence. In addition to motion and size based matching, authors in [25] use a trained siamese network for historical appearance based matching which specifically help in reducing identity switches when tracking. Pair-wise cost calculation in [26] is done using two deep networks where information from spatial attention network (that follows siamese architecture comparing detection bounding boxes and track history) is used in temporal attention network (following Long short-term memory: LSTM architecture) to calculate final pair-wise cost between each detection and track. This dual network is used for tracks and detections that are not matched by the single object tracker (ECO-HC - a hand crafted variant of [27] using Histogram Of gradients (HoG) and Color Names) defined for each track. In [28], authors dynamically initiate sub network for each instance of a person, to predict the next location and use the intersection over union and detection confidence in association cost computation. Authors in [29] uses discriminative appearance learning method for each track using the detection as the positive sample and regions around it as negative samples. They further make use of object size and position based spatio-temporal matching and combine these three measures in a multiplicative manner.

In addition to pair-wise cost based online methods, filter update based tracking is also used in online MOT problem. In [30], object position is predicted in a Gaussian Mixture Probability Hypothesis Density (GM-PHD) filter which is used with a size based location measure in discriminative correlation based appearance cost calculation. In a similar manner Zeyu *et al.* [31] use Monte Carlo PHD filter, where the appearance feature is in the form of dictionary matching defined using RGB color histogram and HoG clustering. Poisson multi-Bernoulli mixture filter is used in [32], where authors use predictions of object 3D co-ordinates that are learned from a deeply trained network. Extended Kalman Filter variant is used by [33] to track object 2D image coordinates, 3D world (using stereo matching and ego-motion calculation) co-ordinates and object size. In certain work [22], Markov Decision Process is used in online tracking framework to update the state (*active, track, lost and inactive*) of the track at each frame using an appearance based template matching, based on dense optical flow matching. Wongun Choi in [15] proposes an energy minimization framework for tracking, where the energy terms are calculated using an appearance model and motion model (using FAST features trajectories measured using optical flow).

Most of the pair-wise methods [14], [16], [21], [22], [29] use Hungarian assignment [12], while some methods [23] use greedy assignment when finding the best match for detections and tracks. In [24] hierarchical association is used based on two different pair-wise matching costs.

Even though deep learning based MOT methods [25], [26] can yield accurate tracking results, these methods in generally are time consuming and require a sizable amount of training data in comparison to traditional hand crafted feature based methods. Some deep learning methods [24] achieve accuracy as well as real time performance, but still require specialized hardware and consume much power to achieve higher running speed. Such methods cannot be used in processing power limited, battery powered embedded systems used in most robotic applications. Some filter update based methods [30], [31] achieve state-of-the-art accuracy without requiring specialized hardware, but their computational time is not high enough for real time applications. Appearance learning based methods [29] can achieve competitive performance in terms of accuracy at a higher computational cost. Thus, simpler methods are required for real time robotic applications that can achieve a good trade off between accuracy and real time performance. In this work, we try to address this challenge. Therefore, we propose to combine multiple matching cost calculation methods that are hand crafted and computationally less expensive to calculate. Specifically, these methods are defined based on object appearance, size and motion. Car tracking methods proposed in [16], [21] use multiple hand crafted features in their framework. However, our method out perform them in terms of accuracy and run time. Authors in [16] use deeply learned feature based matching while [21] uses Chi-Square similarity of the RGB histograms when computing appearance similarity. In the proposed method, color histogram based matching and structure based matching measures are defined when computing appearance similarity. In our method we show that following a grid based histogram matching, can yield a higher accuracy rather than using a single histogram. Hungarian algorithm [12] is used in assigning objects to tracks. Furthermore, we introduce a method to compensate the false negatives arising due to detector failures

for stable tracks. We show that the proposed method can achieve state of the art performance in both human and car tracking with real time running speeds.

III. METHODOLOGY

The proposed tracking method mainly consists of three major components, (a) Dissimilarity Cost Computation, (b) Track Match and Update and (c) Tracking False Negatives for Stable Tracks, which are discussed in detail in this section. Since we are following the tracking by detection framework, for the proposed tracking method we assume that at every frame a set of object bounding boxes along with object class and detection confidence is available.

A. DISSIMILARITY COST COMPUTATION

Dissimilarity cost $dis(i, j)$ between i^{th} detection in the current frame with the j^{th} object in the track list is defined as a weighted sum of four distance measures based on appearance distance $app(i, j)$, structural distance $str(i, j)$, location difference $loc(i, j)$ and size difference $iou(i, j)$ as defined in (1).

$$dis(i,j) = w_{app} \cdot app(i,j) + w_{str} \cdot str(i,j) + w_{loc} \cdot loc(i,j) + w_{iou} \cdot iou(i,j) \quad (1)$$

$w_{app}, w_{str}, w_{loc}$ and w_{iou} are the weights for the contributions from each of the four distance measures. All four distance measures are defined such that they vary between 0 and 1.

1) APPEARANCE BASED DISTANCE

Appearance of an object is an important clue when tracking, as it can be used to recognize the object in different frames as well as to differentiate with other objects. However, the object appearance may be subjected to changes with time due to illumination changes, view-point changes and deformations. Therefore, it is important to define a method to capture or compensate these changes in an appearance feature. To address this problem, in this work we propose a grid based multiple histogram matching method as an appearance feature. Appearance based distance is measured using the correlation between the color histograms of the i^{th} detection and j^{th} track. Histogram matching is done against a maximum of three color histograms extracted from the history of the track. In this work we consider the HSV (hue, saturation, value) color space because it is more robust for illumination changes compared to the RGB (red, green, blue) color space. Hence we use H and S variants for histogram matching. Furthermore, a grid based structure is used when calculating the histogram for each object bounding box. This results in n (number of grid cells per bounding box) number of histograms, which is then concatenated to create a single histogram. Grid based structures have been used to define gradient based keypoint features (known as scale invariant feature transformations: SIFT) from images [34] as well as in generating histogram of oriented gradients (HoG) for human detection [35]. In this work, grid needs to be designed appropriately such that it is not too big nor too small. Implementation parameters are discussed in section IV-B. Since an object can be partially occluded in a given instance, we define an occlusion map for this grid structure to be used when matching histograms to avoid any bias from inaccurate information. Fig. 2 visualize an example on the grid structure and the appearance history of a track.

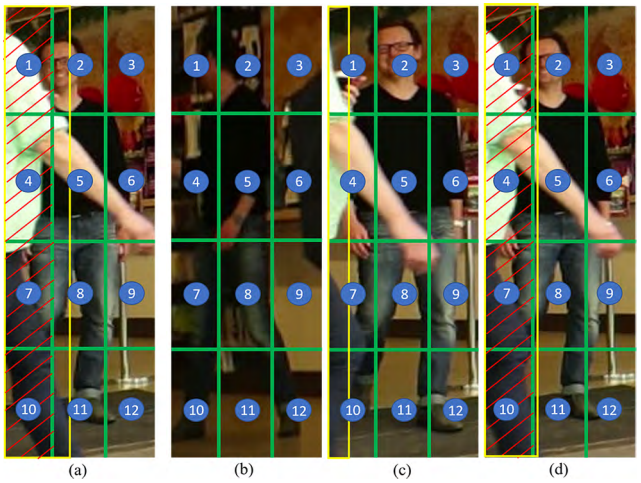

FIGURE 2. Grid structure and appearance based history of a track. (a) current frame matched detection and (b), (c) and (d) are the visualizations of the three appearance history of the matched track kept in memory. Green boxes indicate the grid structure and shaded area by red lines indicate the occlusion. Yellow color indicates the bounding box of the object in front of the track. Grid is 3 × 4. 1^{st}, 4^{th}, 7^{th} and 10^{th} grid cells are occluded due to object in front (yellow bounding box) for (a) and (d). 2^{nd}, 5^{th}, 8^{th} and 11^{th} grid cells are considered non-occluded as yellow bounding box does not cover at least 50% of the grid cell area. No grid cell is occluded for (b) and (c). When appearance matching, 2^{nd} grid cell of (a) is compared with 2^{nd} grid cells of (b), (c) and (d) and average is taken. 1^{st}, 4^{th}, 7^{th} and 10^{th} grid cells are not considered because it is occluded for (a) and all the remaining grid cells are considered in matching.

Since each object is defined in terms of a bounding box, a grid based histogram matching helps to minimize the errors that may otherwise arise due to background. Such situation is presented in Fig. 3. When using single histogram to represent an object, background pixels can sometimes change the histogram giving rise to inaccurate matching, especially if the object is re-appearing after occlusion.

Appearance based distance between i^{th} detection and j^{th} track is defined as in (2).

$$app(i,j) = 1.0 - \frac{\sum_k \sum_n occl(n,i) \cdot occl(n,j)^k \cdot corr(H_{n,i}, H^k_{n,j})}{n \cdot k} \quad (2)$$

where k is the number of concatenated histograms kept in the memory for each track from its history which can be up to a maximum of three. n is the number of cells in the grid. The procedure followed in updating histograms for a track is discussed in section III-B. $occl(n, i)$ and $occl(n, j)^k$ are the occlusion indicators for the n^{th} grid cell of the i^{th} detection and n^{th} grid cell of the j^{th} track of the k^{th} history map, respectively. $corr(H_{n,i}, H^k_{n,j})$ is the correlation between the

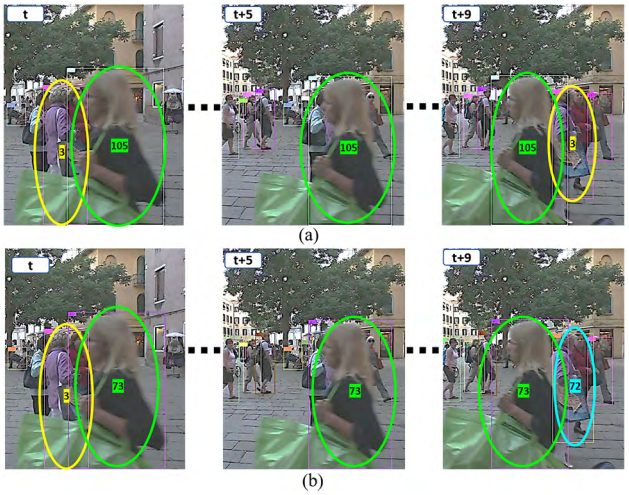

FIGURE 3. Grid based histogram matching against single histogram matching. (a) grid-based concatenated histogram based matching and (b) single histogram based matching. In (a) identity of the track ID 3 is occluded by the person walking in front (track ID 105) is kept successfully when matching with grid-based concatenated histogram. In (b) track ID 3 is falsely matched with tack ID 72 after occlusion. Track ID is magnified for better visualization.

histogram of the n^{th} grid cell of the i^{th} detection $H_{n,i}$ and n^{th} grid cell of the j^{th} track of the k^{th} history map $H^k_{n,j}$. Correlation indicates the similarity between two histograms and output a value between 0 and 1. Hence, we divide the sum of histogram correlations by total number of histograms ($n*k$) and take one minus that as the appearance distance. Correlation between any two histograms $corr(H_a, H_b)$ is calculated based on (3).

$$corr(H_a, H_b) = \frac{\sum_I (H_a(I) - \bar{H_a})(H_b(I) - \bar{H_b})}{\sqrt{\sum_I (H_a(I) - \bar{H_a})^2 \sum_I (H_b(I) - \bar{H_b})^2}} \quad (3)$$

where,

$$\bar{H_c} = \frac{1}{N} \sum_J H_c(J) \quad (4)$$

in which N is the total number of bins in the histogram.

Occlusion parameter *occl* for each grid cell is either 1 or 0 depending on the level of occlusion for that particular grid cell. Level of occlusion is defined based on the bounding box overlap. If any two or more bounding boxes overlap with each other, the bounding box belonging to the object that has the highest detection confidence is considered to be in front and the overlapping areas with other bounding boxes are considered to be occluded areas for those bounding boxes. Then in a given grid cell, if more than 50% of the pixels are occluded in such a way, the occlusion parameter for that grid cell is set to 0, otherwise set to 1 as presented in Fig. 2.

The appearance cost is defined as an average based on k concatenated histograms in track's history rather than considering the maximum response from one. Reason for such measure is to avoid track drifting based on the most recent tracked detection (mostly when the track is partially occluded

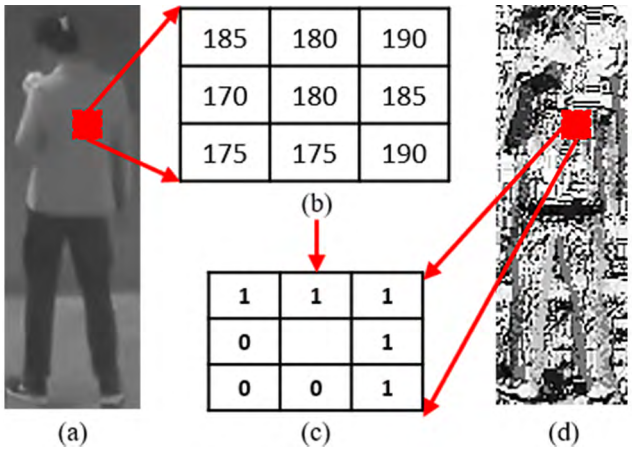

FIGURE 4. Linear Binary Pattern (LBP). (a) Gray scale image. (b) Zoom-in view of the area of the red box of (a). (c) LBP of (b). (d) LBP of full image (a).

by another object and to avoid identity switch) as one of the concatenated histograms in track memory is always the concatenated histogram of the immediate history of the track as explained in section III-B. Also refer to Fig. 2 for an example on an object in front of the track. This is expressed in (5) below, where $M_{n,i}$ is the total number of pixels in the n^{th} grid cell and $m_{occl,n,i}$ is the number of pixels occluded in that cell.

$$occl(n, i) = \begin{cases} 1; & if\ m_{occl,n,i} > 0.5 \times M_{n,i} \\ 0; & otherwise \end{cases} \quad (5)$$

2) STRUCTURE BASED DISTANCE

Structure is another clue that can be used in recognition task. Linear Binary Pattern (LBP) [36] is a good feature that can capture structural information from an image and is a robust feature for illumination variances. LBP outputs a binary code word for each pixel by comparing pixels values in a 3x3 neighborhood with the center pixel value. Such example is presented in Fig. 4. LBP image is a collection of 8 bit binary code words out of all possible code words (in 3x3 neighborhood $2^{9-1} = 256$ code words), that can be represented as a histogram. This is known as the LBP Histogram (LPBH). In [37] Ahonen et al. used LBPH in a grid structure for face recognition task and achieved the best results at the time. In grid structure LBPH captures the shape of the nose, eyes, mouth and other shapes in the face which can be then used for recognition. In object level, LPBH may not be a good feature to differentiate one another as a given object class can have very similar shape. But when considering a grid based structure and concatenated LBPH matching, structural information can be used to find the best candidate. Unlike in face domain, at object level structure may not be a reliable feature for recognition but the object level structure can be a good feature to measure the correlation between two structures in adjacent frames as the structure of an object in not expected to change drastically in adjacent frames of a video. Hence, we use grid based LBPHs

to define a distance measure in this tracking task. The same grid structure described in the section III-A.1 is used here was well. Unlike during the appearance based matching, we only match the concatenated LBPH of the detection with the most recent concatenated LBPH of the track. The structure based distance measure between i^{th} detection and j^{th} track, $str(i, j)$ is as defined in (6).

$$str(i,j) = 1.0 - \frac{\sum_n corr(LBPH_{n,i}, LBPH_{n,j})}{n} \tag{6}$$

where $corr(LBPH_{n,i}, LBPH_{n,j})$ is the correlation between the LBPH of the n^{th} grid cell of the i^{th} detection $LBPH_{n,i}$ and n^{th} grid cell of the j^{th} track $LBPH_{n,j}$.

3) MOTION BASED DISTANCE

When tracking an object in adjacent frames, its motion helps to predict the position of the object in the next frame. We define the difference between the predicted position $pred(x, y)$ and the measured position $det(x, y)$ as the motion based distance measure as in (7). L2 norm is used. x and y are the center positions of the object bounding box expressed in 2D image coordinates. n_{dst} is the value used to normalize the distance measure.

$$loc(i,j) = \min\left(1\ ,\ \frac{\parallel det(x,y) - pred(x,y) \parallel_2}{n_{dst}}\right) \tag{7}$$

Predicted position $pred(x, y)$ is calculated based on the object bounding box information from its track history. The predictions for the bounding box center (x, y), width w and height h is generated using a weighted average of it's previous five frames information. We use five frames because it is large enough to get good average for predictions and small enough not to drift because of old information. We use weights to give a higher weightage to the information from the most recent track assignments in the history. Predicted bounding box bb_j^p of the j_{th} object in existing tracks is calculated based on (8) below, where t is number of frames the j^{th} track is matched with a detection (i.e. bb_j^t bounding box contains the most recent matched detection bounding information) and x, y, w and h are bounding box parameters. m is the consecutive number of frames that track is undetected.

$$bb_j^p = \begin{cases} x_{bb_j^t} + \Delta_x(1+m); \\ y_{bb_j^t} + \Delta_y(1+m); \\ w_{bb_j^t} + \Delta_w(1+m); \\ h_{bb_j^t} + \Delta_h(1+m); \end{cases} \tag{8}$$

where $\Delta_z : z \in \left[x, y, w, h\right]$ is defined as in (9).

$$\Delta_z = \frac{\sum_{a=2}^{5}\left[(a-1)(z_{t+a-5} - z_{t+a-6})\right]}{\sum_{a=2}^{5}(a-1)} \tag{9}$$

Predicted position $pred(x, y)$ is defined as in (10) and width of the most recent bounding box $w_{bb_j^t}$ is used as n_{dst}.

$$pred(x,y) = \begin{cases} (x,y)_{bb_j^t}; & if\ t < 5 \\ (x,y)_{bb_j^p}; & otherwise \end{cases} \tag{10}$$

4) SIZE BASED DISTANCE

Size based dissimilarity is calculated based on the object localization information (object bounding box) of each detection compared to the predicted localization information from object track history. Specifically, intersection over union (IoU) between the detection bounding box $bb_{i,det}$ of the i_{th} detection and the predicted bounding box $bb_{j,pred}$ of the j_{th} track is considered. IoU is inversely proportional to the dissimilarity between two bounding boxes and the IoU will directly provide the similarity value normalized to $0-1$. Therefore, size based dissimilarity is defined below as in (11).

$$iou(i,j) = 1 - \frac{bb_{i,det} \cap bb_{j,pred}}{bb_{i,det} \cup bb_{j,pred}} \tag{11}$$

IoU is measured considering both the location and size of the bounding boxes in comparison, thus is a good measure of the location similarity of bounding boxes. Predicted bounding box $bb_{j,pred}$ for the j^{th} track is as below in (12), where bb_j^t and bb_j^p is as in (8).

$$bb_{j,pred} = \begin{cases} bb_j^t; & if\ t < 5 \\ bb_j^p; & otherwise \end{cases} \tag{12}$$

B. TRACK MATCH AND UPDATE

Once all four dissimilarity measurements are calculated, overall dissimilarity values $dis(i, j)$ is calculated as in (1) for each i^{th} detection in the current frame against each j^{th} track in the memory. Therefore, this dissimilarity matrix is used in the Hungarian algorithm [12] to assign each of the object detection in the current frame to the best matching track. However, this association is done only if the dissimilarity between the matched detection and track is below a certain threshold th_{dis}, otherwise the detected object is initiated as a new track after tracking for false negatives detailed in the section III-C.

In the detection and track association stage, object information in the list of tracks is updated with the current detection's localization co-ordinates, LBPHs and detection confidence. Appearance based histograms and occlusion information are updated in a different manner. For each track three most relevant concatenated appearance histogram and occlusion information are stored. Firstly, from the already existing three most relevant concatenated histograms the most occluded one is removed. If the level of occlusion is similar, the oldest concatenated histogram is removed. Then the current matched detection's concatenated appearance histogram and occlusion information is stored for the track.

C. TRACKING FALSE NEGATIVES FOR STABLE TRACKS

One of the applications of a tracking algorithm following tracking-by-detection framework is to identify and maintain the track of an object that goes undetected in some frame/frames (as a result of a false negative of the detection algorithm), of a stable track. In this section we discuss the method we propose to address this challenge. Fig. 5 visualize

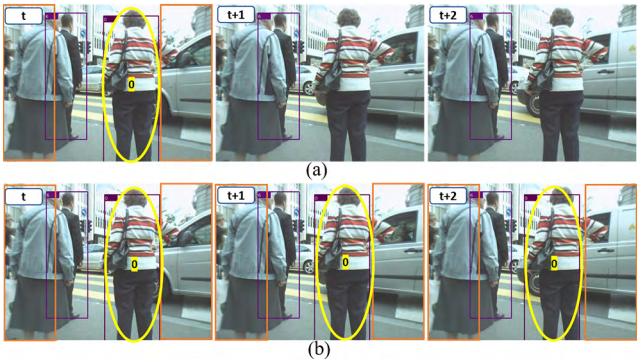

FIGURE 5. Maintaining the track in case of a false negative in detection. (a) object with ID 0 is undetected in adjacent frames (t+1 and t+2) and (b) track ID maintained based on the track history. Track ID is magnified for better visualization. Orange boxes indicate the exit regions for track ID 0.

an example on such missed detection successfully addressed by the proposed method.

In this work we define a track is stale if it has been tracked in at least five previous frames, as it is the number of frames required to predict the localization information for a track based on (8) and (9). Tracking for false negatives is initiated after the track matching and updating step described in the section III-B but before initiating new tracks for unmatched detections and invalidating tracks as shown in the flow chart in Fig. 1.

After the Hungarian assignment there are three outputs, (a) i^{th} detection matched with j^{th} track, (b) unmatched detection and (c) unmatched track. For each of these unmatched tracks, size based dissimilarity measure is calculated against all the unmatched detections, according to (11) where the predicted bounding box $bb_{j,pred}$ for the unmatched track is bb_j^p as per the (8) and (9). If none of the above size based dissimilarity measures, is not less than a threshold th_{iou} (0.5 is used as the threshold in this study during the experiments), we calculate the appearance based dissimilarity according to the (2) with a new detection defined using the predicted bounding box $bb_{j,pred}$ for that track. If the appearance based dissimilarity is less than a threshold th_{app} (0.6 is used as the threshold in this study during the experiments) the predicted bounding box $bb_{j,pred}$ for that track is considered as its current frame match. However, if the predicted bounding box is out of the frame, the corresponding track is immediately marked as an invalid track. Additionally if the predicted bounding box is fully within the exit regions as indicated in Fig. 5, it is not considered as a match. Width of the exit regions are dynamically defined for each track as the width of the track's last matched bounding box width.

In case of at least one detection has a dissimilarity measure less than th_{iou}, these tracks and detections are considered in Hungarian assignment and matched with each other. Association is done only if the dissimilarity between the matched detection and track is below the same threshold th_{iou}, otherwise the detected object is initiated as a new track.

After updating the track list with the current matched detection data and tracking for false negatives, the track list is updated to invalidate certain tracks which have not being tracked (i.e. matching object not found) for a predefined consecutive number of frames.

IV. RESULTS AND DISCUSSION

The proposed tracking framework is implemented using C++ and all the test cases presented in this section is done in a desktop with Intel Xeon(R) CPU E5-1630 v3 with 3.70 GHz 8 processors. Even though there are 8 processors, code is not optimized for parallel processing.

MOT16 [10] is a popular benchmark for human tracking while KITTI [11] vision benchmark suite is a famous benchmark for autonomous driving based applications. We tested the proposed tracking methodology in both these data-sets. MOT17[1] includes all sequences of MOT16 with a new, more accurate ground truth with three sets of detections for each sequence. Hence, we use MOT17 benchmark for human tracking evaluation. MOT17 tracking data-set consists of 21 training sequences and 21 test sequences, where in each of these sets, 2 sequences consist of images with 640 × 480 resolution while the remaining 19 sequences contain images of 1920 × 1080 resolution. MOT17 training set has a total of 15,948 frames with an average of 21.1 detections per frame, while the test set contain a total of 17,757 frames with an average of 31.8 detections per frame. KITTI tracking data-set consists of 21 training sequences and 29 test sequences, where 31 sequences contains images of size 1242 × 375 while other sequences contain images of similar resolution (i.e. $12xx \times 37x$ resolutions). KITTI training set has a total of 8,008 frames with an average of 3.8 detections per frame and the test set contain a total of 11,095 frames with an average of 3.5 detections per frame. Since the proposed tracking method does not require any training on tracking data, both training and test data-sets have been used as test beds in our experiments. The evaluation codes by MOT17 and KITTI benchmarks were used, which are based on CLEARMOT [38] and Mostly Tracked (MT: % of ground truth trajectories which are covered by tracker output for more than 80% in length) − Mostly Lost (ML: % of ground truth trajectories which are covered by tracker output for less than 20% in length) [39] metrics. CLEARMOT [38] includes total Identity Switches (IDS: The total of number of times that a tracked trajectory changes its matched ground truth identity), Multiple Object Tracking Accuracy (MOTA), Multiple Object Tracking Precision (MOTP) and average run-time per frame excluding detection time (Time). Here, $MOTA = 1 - \frac{\sum_t (fn_t + fp_t + IDS_t)}{\sum_t g_t}$ and $MOTP = \frac{\sum_{i,t} d_t^i}{\sum_t c_t}$, where t is the frame t, fn is false negatives, fp is false positives, g is ground truth detections, c_t is correct matches found at frame t and d_t^i is the distance between predicted detection and ground truth detection for each correct match which is taken as the intersection of union between the two bounding boxes.

[1] MOT17: https://motchallenge.net/data/MOT17/

A. DETECTIONS

In the proposed method we are following the tracking-by-detection framework. For the MOT17 data, the benchmark has made detections available for each sequence. In fact it has made available detections from three human detectors namely, DPM [40], F-RCNN [41] and SDP [42]. In total there are 14 unique sequences and with three detection results for each make it 42 sequences. KITTI benchmark does not provide any detection results. Therefore, we used bounding box results from RRC detector [19].

Bonding boxes from DPM [40] human detector has higher number of false positives in general compared to other detectors. Therefore, as a pre-processing step we introduce a filtering criteria for DPM based detections by calculating a threshold for detection confidence for each sequence. At each frame, we search for the detections that overlaps with other detections. If the overlap is greater than 50% of the detection, we compare the detection confidence of the detection in consideration against the most overlapped detection's confidence and invalidate the bonding box with the lower confidence. This is repeated for the whole sequence and filter some detections. Then based on these invalidated detection list we define the threshold for detection confidence as the average of all these invalidated detection confidences. This is done for each sequence separately because of the large differences in sequences in terms of view-point, dynamic & static camera, illuminations differences and etc. Then all the detections which has a lower confidence than the threshold is marked as invalid and not used in tracking.

B. ABLATION STUDY AND PARAMETERS TUNING

Ablation study is conducted to understand the contribution from each component defined in (1) and the contribution from tracking false negative part described in section III-C.

We conducted all these experiments in MOT17 and KITTI training data. First, we evaluate the contribution from each component in (1) where we show the results in Table 1 for tracking accuracy when the tracks and detections matching is done only using each component in rows *app*, *str*, *loc* and *iou*. When comparing MOTA values it is evident that each component contribute almost equally for tracking accuracy. Therefore, we define the weights in (1) to be equal, i.e. $w_{app} = w_{str} = w_{loc} = w_{iou} = 0.25$. Hence, we maintain $dis(i, j)$ in (1) between 0 and 1.

Performance based on combining different components are analyzed. Since the main feature is based on appearance, we analyze the combinations that include *app* component. Results in the *dis* row is when all the four components are combined according to (1) (i.e. $app + str + loc + iou$). Even though the MOTA is similar to the nearest decimal, IDS is lowest when all the four components are combined. The MASS row shows the results when *'tracking false negatives for stable tracks'* is included on top of *dis*. The MOTA is increased approximately by 0.8 % when *'false negative tracking for stable tracks'* is incorporated for the MOT17 data.

TABLE 1. Ablation study in MOT17 and KITTI tracking training set.

	MOTA	MOTP	IDS	MT	ML
MOT17					
app	48.3%	83.4%	4,512	22.8%	32.5%
str	48.2%	83.4%	4,550	22.6%	32.5%
loc	48.3%	83.4%	4,295	22.5%	32.5%
iou	48.3%	83.4%	4,439	22.6%	32.5%
$app + str$	48.5%	83.4%	3,737	22.6%	32.5%
$app + loc$	48.7%	83.4%	3,061	22.5%	32.5%
$app + iou$	48.7%	83.4%	3,015	22.6%	32.5%
$app + str + loc$	48.6%	83.4%	3,194	22.6%	32.4%
$app + iou + loc$	48.6%	83.4%	3,235	22.5%	32.4%
$app + iou + str$	48.7%	83.4%	3,150	22.6%	32.5%
dis	48.7%	83.4%	2,905	22.6%	32.5%
MASS	49.5%	83.1%	2,808	23.8%	31.3%
KITTI					
app	88.3%	89.9%	994	87.8%	1.6%
str	91.7%	89.9%	167	87.8%	1.6%
loc	91.0%	89.9%	326	87.8%	1.6%
iou	91.2%	89.9%	299	87.8%	1.6%
$app + str$	91.4%	89.9%	234	87.8%	1.6%
$app + loc$	91.6%	89.9%	181	87.8%	1.6%
$app + iou$	91.7%	89.9%	180	87.8%	1.6%
$app + str + loc$	91.9%	89.9%	124	87.8%	1.6%
$app + iou + loc$	91.5%	89.8%	212	87.8%	1.6%
$app + iou + str$	91.9%	89.9%	118	87.8%	1.6%
dis	91.9%	89.9%	113	87.8%	1.6%
MASS	92.0%	89.8%	113	88.3%	1.6%

Unlike the human detectors in MOT17, false negatives from the car detector is very low, hence the improvement for KITTI data is not that considerable (about 0.1 %) by incorporating false negative tracking. When considering the results of each individual components for KITTI data, it can be seen that appearance feature *app* based tracking performance is low compared to other individual components due to higher identity switches. When tracking humans, appearance is an important factor since most of the time the clothes have different colors and combination of colors is even more discriminative (i.e. two persons may be wearing black and white color clothes, but one may be using black for the top and the other for the bottom, which can be used to differentiate them). Since the proposed appearance matching is based on grid based color information, different color combinations are captured well. But in the case of cars, they have the same color everywhere which makes the proposed appearance based tracking method has a higher probability for error compared to human tracking. In Table 2, evaluation results for test data is shown as displayed in the benchmarks.

In the proposed tracking framework, after the Hungarian assignment, detection is assigned to the matched track if the dissimilarity cost $dis(i, j)$ is below a certain threshold th_{dis}. The graph in the Fig. 6 shows how MOTA varies with changing th_{dis} for MOT17 and KITTI tracking training data. Based on this, $th_{dis} = 0.75$ is selected for the experiments. Furthermore, number of grid cells n in the appearance feature matching introduced in section III-A.1 has an impact on the accuracy of the proposed tracking framework. It is visualized in the same graph and we selected $n = 12$ as the configuration in this work. Grid cell division is defined based on the object

TABLE 2. Results in MOT17 and KITTI tracking test set as displayed in the benchmark online (accessed in April 2019).

	MOTA	MOTP	MT	ML	IDS	Time	Environment
MOT 17							
MOTDT17 [24]	**50.9 %**	76.6 %	17.5 %	35.7 %	2,474	**18.3 Hz**	3 GHz, GTX 1060 (Caffe)
MTDF17 [30]	49.6 %	75.5 %	18.9 %	**33.1 %**	5,567	1.2 Hz	3.5 GHz, GTX 1060
HAM_SADF17 [25]	48.3 %	**77.2 %**	17.1 %	41.7 %	**1,871**	5 Hz	3.7 GHz, TITAN X
DMAN [26]	48.2 %	75.7 %	**19.3 %**	38.3 %	2,194	0.3 Hz	2.4 GHz (Matlab+Tensoflow)
AM_ADM17 [29]	48.1 %	76.7 %	13.4 %	39.7 %	2,214	5.7 Hz	3.4 GHz
PHD_GSDL17 [31]	48.0 %	**77.2 %**	17.1 %	35.6 %	3,998	6.7 Hz	3.5 (Matlab) GHz
MASS (ours)	46.9 %	76.1 %	16.9 %	36.3 %	4,478	17.1 Hz	3.7 GHz (C++)
KITTI							
MOTBeyondPixels [16]	84.24 %	**85.73 %**	73.23 %	**2.77 %**	468	0.3 s	2.5 GHz (C/C++)
IMMDP [22]	83.04 %	82.74 %	60.62 %	11.38 %	172	0.19 s	3.5 GHz (Matlab + C/C++)
3D-CNN/PMBM [32]	80.39 %	81.26 %	62.77 %	6.15 %	**121**	**0.01 s**	3 GHz (Matlab)
extraCK [21]	79.99 %	82.46 %	62.15 %	5.54 %	343	0.03 s	2.5 GHz (Python)
MASS (ours)	**85.04 %**	85.53 %	**74.31 %**	**2.77 %**	301	**0.01 s**	3.7 GHz (C/C++)

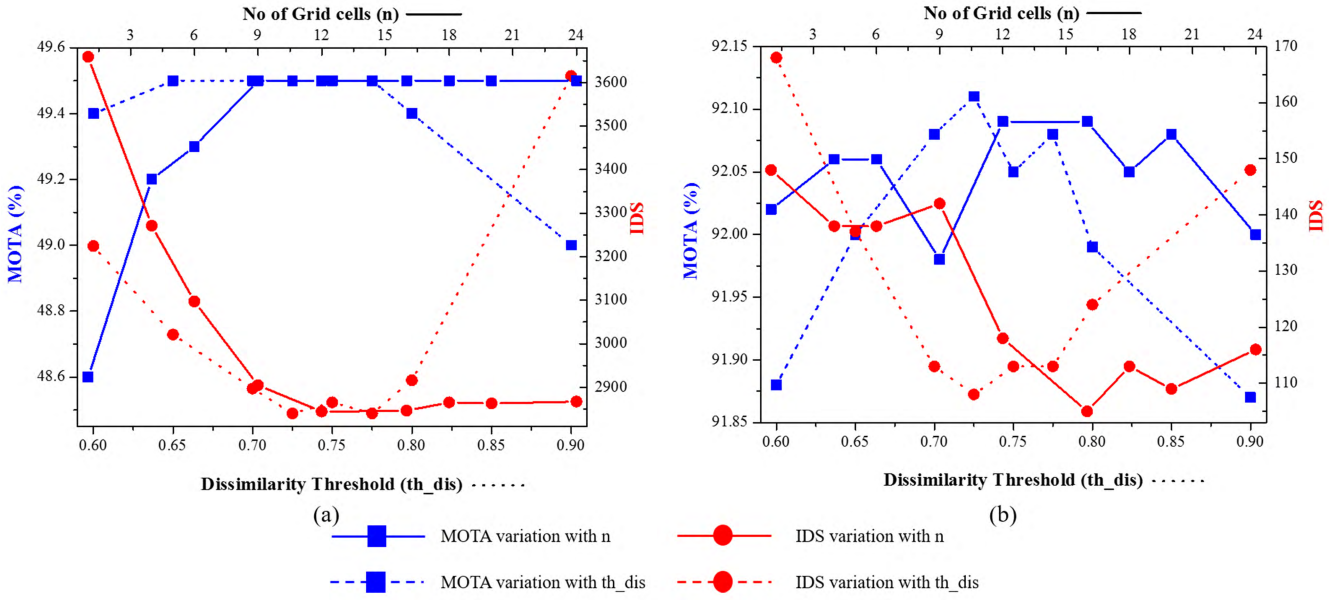

FIGURE 6. Comparison of Multiple Object Tracking Accuracy (MOTA) and Identity Switches (IDS) variations for different dissimilarity thresholds (th_{dis}) and number of grid cells (n). (a) for MOT17 Training Data and (b) for KITTI Training Data. Blue color represent MOTA and red color represent ID Switches. MOTA and IDS values based on dissimilarity threshold th_{dis} are in dotted lines while solid lines are based on number of grid cells n.

class. Human object has a greater height than the width where as for car object it is the opposite. Therefore, when dividing $n = 12$ grid cells, (3×4) is assigned for human class and (4×3) is assigned for car class. We are using the HSV color space for color histogram where we use 15 bins for hue (H) and 16 bins for saturation (S). The average processing time per frame is 35 ms for MOT17 tracking training data and 5 ms for KITTI tracking training data.

C. COMPARISON WITH OTHERS

In this section we compare the proposed tracking method with other state-of-the-art methods listed in the two benchmarks. Comparison is made between the online methods that are published. Summary of the performance of the selected trackers are as shown in Table 2. The proposed method is compared with the best tracking frameworks in the KITTI [11] and MOT17 [10] benchmarks that are published and are vision based online multiple object tracking methods. Tracking performance with the test data sets are compared in the benchmarks. Test data is more challenging than the train data in both benchmarks, which is why there is a performance drop compared to results on training data stated in Table 1.

Our method is ranked top in the KITTI benchmark for online tracking methods that are published and third place among all online methods, in terms of MOTA. MOTBeyondPixels [16], reports that it uses inputs from two object detectors and is the next best to ours. In KITTI benchmark the lowest amount that can be entered for time is 0.01 s. In 3D-CNN/PMBM [32] authors record a running time of 73 fps for the test set while our method can achieve 150+ fps. However, it should be noted that 3D-CNN/PMBM [32] is based on Matlab.

In MOT17, out of all the published online tracking methods, the proposed tracker is ranked in seventh place at the

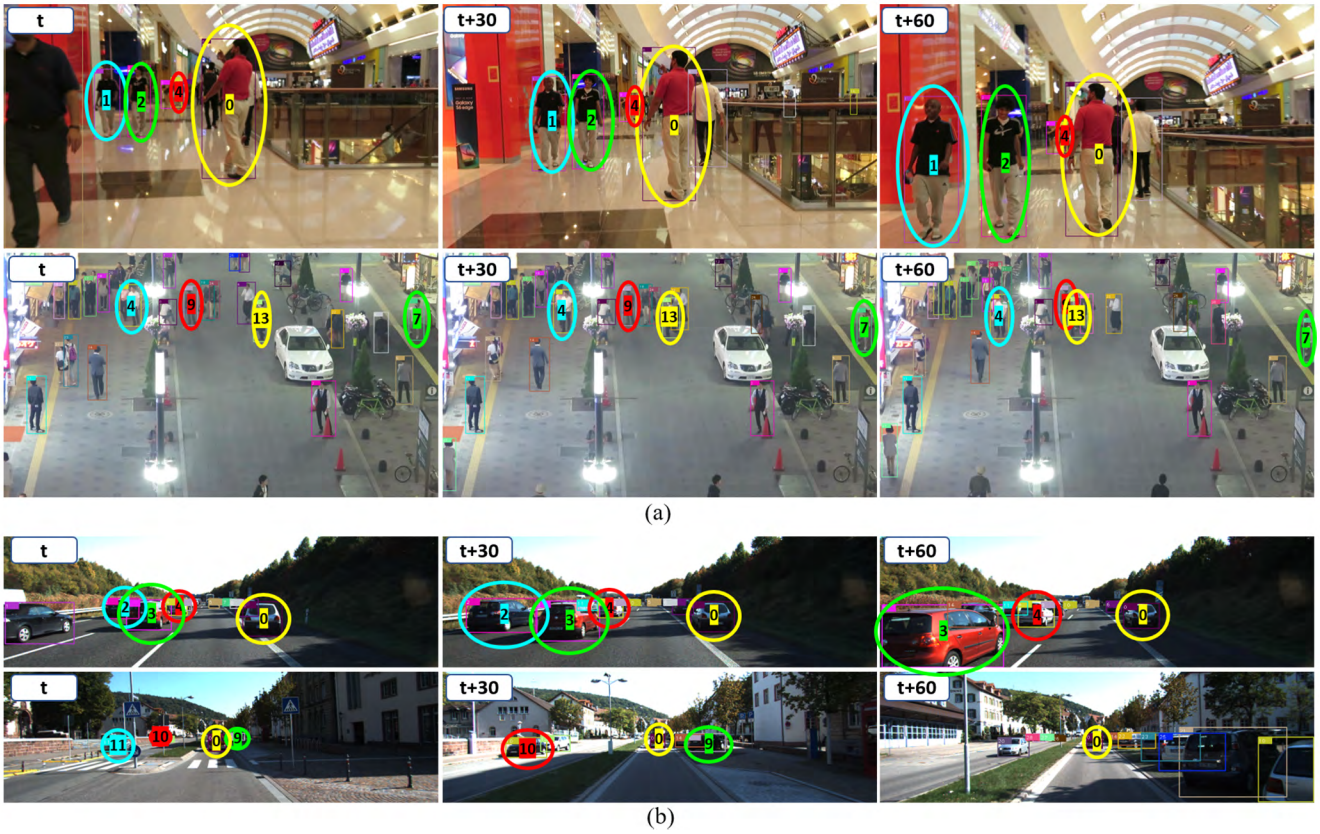

FIGURE 7. Tracking results on few frames. (a) human tracking results from MOT17 training data and (b) car tracking results from KITTI training data. At each sequence four tracks are highlighted using yellow, green, blue and green colors. These identities are uniquely tracked for more than 60 frames.

time of results submission. However, it recorded the second best running speed and is nearly three times the speed of the third best method PHD_GSDL17 [31] even though PHD_GSDL17 [31] is implemented in Matlab. Out of the best six methods, HAM_SADF17 [25] and DMAN [26] pair-wise cost calculations are based on deeply leaned neural networks while MOTDT17 [24] appearance cost is based on deeply learned person re-identification cost. MTDF17 [30] uses detection information from two object detectors in its detection stage. Compared to AM_ADM17 [29] and PHD_GSDL17 [31] the proposed method has lower MOTA but higher running speeds. The proposed method is written using C++ as it was to be implemented in an embedded system for a real time application. The proposed method achieves a good compromise between the speed and the accuracy, making it suitable to be used in real time robotics application with no requirement for specialized hardware for pedestrian tracking. Tracking results on few random frames is visualized in Fig. 7, where you can see the track ID is consistent across the frames.

The running time of the proposed method depend on the complexity of the scene rather than the frame size. MOT17 training data-set has an average of 21.1 detections per frame, while the test set has an average of 31.8 detections per frame. Running time for MOT17 training set is 35 ms while for test set it is 55 ms per frame. This reflects

TABLE 3. Comparison of results on different detection methods on MOT17-3rd sequence.

	MOTA	MOTP	IDS
MOT17-03 DPM [40]			
MOTDT17 [24]	55.5 %	71.3 %	**364**
MTDF17 [30]	**56.8 %**	72.9 %	992
MASS (ours)	45.1 %	**74.7 %**	602
MOT17-03 F-RCNN [41]			
MOTDT17 [24]	58.7 %	77.3 %	**220**
MTDF17 [30]	**59.9 %**	76.2 %	683
MASS (ours)	58.0 %	**77.6 %**	347
MOT17-03 SDP [42]			
MOTDT17 [24]	73.2 %	**78.5 %**	**361**
MTDF17 [30]	72.5 %	78.3 %	728
MASS (ours)	**73.6 %**	77.1 %	636

the average detections per frame in each set. Compared to MOT17, KITTI [11] training and test data has similar average detections per frame (3.5 for test set and 3.8 for training set) and result in similar running speeds (5 ms per frame). When comparing MOT17 and KITTI, running speeds are higher in KITTI data because the average number of detections per frame is lower than that of MOT17. The proposed tracking algorithm depends on number of detentions in the current frame plus the number of tracks in the track list, which justify the observed behavior.

Tracking performance of the methods following tracking-by-detection framework is affected by the detection method

used. In Table 3, tracking performance of the proposed method and the best two methods in Table 2 is presented considering the third sequence of the MOT17 benchmark. All the three methods in Table 3 perform best when SDP [42] detection results are used, followed by when F-RCNN [41] and DPM [40] detection results are used. When SDP [42] is used as the detector the proposed method achieve the best MOTA. The proposed method performs competitively when F-RCNN [41] is used as the detector but show poor results when DPM [40] detection results are used in comparison with other two methods. As explained in section IV-A DPM detections contain large number of false positives and our method does not handle these as well as other methods.

V. CONCLUSION

In this paper we have presented an efficient framework for multiple object tracking problem that runs online and achieves 150+ fps for KITTI data and 28.5 fps and 18.3 fps for MOT17training and test data respectively. Thus, the proposed method is highly suitable for real time applications and the results show that the proposed method achieve state-of-the-art performance for both human tracking and car tracking. The running time increases with the complexity of the scene (i.e. on the number of objects present in the frame). The proposed tracker is based on combination of grid based color histogram matching, grid based LBPHs matching, predicted object motion matching and predicted size based matching. Furthermore, in this work we have proposed a false negative tracking to compensate errors from the object detector. As with all tracking-by-detection methods, tracking performance depends on the accuracy of the detection results, where the proposed method show very competitive results when there are less number of false positives in detections. Furthermore, when comparing the results in Table 2, IDS are higher in the proposed method. Therefore, including a false positive removal mechanism as well as post processing method to reduce IDS can be future directions to further improve the tracking accuracy.

REFERENCES

[1] A. Adam, E. Rivlin, and I. Shimshoni, "Robust fragments-based tracking using the integral histogram," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, vol. 1, Jun. 2006, pp. 798–805.

[2] Y. Wu, J. Lim, and M.-H. Yang, "Online object tracking: A benchmark," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2013, pp. 2411–2418.

[3] Z. Kalal, K. Mikolajczyk, and J. Matas, "Tracking-learning-detection," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 34, no. 7, pp. 1409–1422, Jul. 2012.

[4] D. S. Bolme, J. R. Beveridge, B. A. Draper, and Y. M. Lui, "Visual object tracking using adaptive correlation filters," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2010, pp. 2544–2550.

[5] J. F. Henriques, R. Caseiro, P. Martins, and J. Batista, "High-speed tracking with kernelized correlation filters," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 37, no. 3, pp. 583–596, Mar. 2015.

[6] B. Chen, P. Li, C. Sun, D. Wang, G. Yang, and H. Lu, "Multi attention module for visual tracking," *Pattern Recognit.*, vol. 87, pp. 80–93, Mar. 2019.

[7] D. Held, S. Thrun, and S. Savarese, "Learning to track at 100 FPS with deep regression networks," in *Computer Vision—ECCV*, B. Leibe, J. Matas, N. Sebe, and M. Welling, Eds. Cham, Switzerland: Springer, 2016, pp. 749–765.

[8] 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 *Computer Vision—ECCV Workshops*, G. Hua and H. Jégou, Eds. Cham, Switzerland: Springer, 2016, pp. 68–83.

[9] S. Tang, M. Andriluka, B. Andres, and B. Schiele, "Multiple people tracking by lifted multicut and person re-identification," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jul. 2017, pp. 3539–3548.

[10] A. Milan, L. Leal-Taixé, I. Reid, S. Roth, and K. Schindler, "MOT16: A benchmark for multi-object tracking," Mar. 2016, *arXiv:1603.00831*. [Online]. Available: http://arxiv.org/abs/1603.00831

[11] A. Geiger, P. Lenz, and R. Urtasun, "Are we ready for autonomous driving? The KITTI vision benchmark suite," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2012, pp. 3354–3361.

[12] H. W. Kuhn, "The Hungarian method for the assignment problem," *Naval Res. Logistics Quart.*, vol. 2, nos. 1–2, pp. 83–97, Mar. 1955.

[13] R. Henschel, L. Leal-Taixé, D. Cremers, and B. Rosenhahn, "Fusion of head and full-body detectors for multi-object tracking," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. Workshops (CVPRW)*, Jun. 2018, pp. 1428–1437.

[14] A. Sadeghian, A. Alahi, and S. Savarese, "Tracking the untrackable: Learning to track multiple cues with long-term dependencies," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Venice, Italy, Oct. 2017, pp. 300–311. doi: 10.1109/ICCV.2017.41.

[15] W. Choi, "Near-online multi-target tracking with aggregated local flow descriptor," in *Proc. IEEE Int. Conf. Comput. Vis.*, Dec. 2015, pp. 3029–3037.

[16] S. Sharma, J. A. Ansari, J. K. Murthy, and K. M. Krishna, "Beyond pixels: Leveraging geometry and shape cues for online multi-object tracking," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, May 2018, pp. 3508–3515.

[17] J. H. Yoon, C.-R. Lee, M.-H. Yang, and K.-J. Yoon, "Online multi-object tracking via structural constraint event aggregation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2016, pp. 1392–1400.

[18] C. Kim, F. Li, A. Ciptadi, and J. M. Rehg, "Multiple hypothesis tracking revisited," in *Proc. IEEE Int. Conf. Comput. Vis.*, Dec. 2015, pp. 4696–4704.

[19] J. Ren, X. Chen, J. Liu, W. Sun, J. Pang, Q. Yan, Y.-W. Tai, and L. Xu, "Accurate single stage detector using recurrent rolling convolution," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2017, pp. 5420–5428.

[20] Y. Xiang, W. Choi, Y. Lin, and S. Savarese, "Subcategory-aware convolutional neural networks for object proposals and detection," in *Proc. IEEE Winter Conf. Appl. Comput. Vis. (WACV)*, Mar. 2017, pp. 924–933.

[21] G. Gündüz and T. Acarman, "A lightweight online multiple object vehicle tracking method," in *Proc. IEEE Intell. Vehicles Symp. (IV)*, Jun. 2018, pp. 427–432.

[22] Y. Xiang, A. Alahi, and S. Savarese, "Learning to track: Online multi-object tracking by decision making," in *Proc. IEEE Int. Conf. Comput. Vis.*, Dec. 2015, pp. 4705–4713.

[23] K. Fang, Y. Xiang, X. Li, and S. Savarese, "Recurrent autoregressive networks for online multi-object tracking," in *Proc. IEEE Winter Conf. Appl. Comput. Vis.*, Mar. 2018, pp. 466–475.

[24] L. Chen, H. Ai, Z. Zhuang, and C. Shang, "Real-time multiple people tracking with deeply learned candidate selection and person re-identification," in *Proc. IEEE Int. Conf. Multimedia Expo (ICME)*, Jul. 2018, pp. 1–6.

[25] Y.-C. Yoon, A. Boragule, Y.-M. Song, K. Yoon, and M. Jeon, "Online multi-object tracking with historical appearance matching and scene adaptive detection filtering," in *Proc. 15th IEEE Int. Conf. Adv. Video Signal Based Surveill. (AVSS)*, Nov. 2018, pp. 1–6.

[26] J. Zhu, H. Yang, N. Liu, M. Kim, W. Zhang, and M.-H. Yang, "Online multi-object tracking with dual matching attention networks," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, 2018, pp. 366–382.

[27] M. Danelljan, G. Bhat, F. S. Khan, and M. Felsberg, "ECO: Efficient convolution operators for tracking," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jul. 2017, pp. 6638–6646.

[28] H. Wu, Y. Hu, K. Wang, H. Li, L. Nie, and H. Cheng, "Instance-aware representation learning and association for online multi-person tracking," *Pattern Recognit.*, vol. 94, pp. 25–34, Oct. 2019.

[29] S.-H. Lee, M.-Y. Kim, and S.-H. Bae, "Learning discriminative appearance models for online multi-object tracking with appearance discriminability measures," *IEEE Access*, vol. 6, pp. 67316–67328, 2018.

[30] Z. Fu, F. Angelini, J. Chambers, and S. M. Naqvi, "Multi-level cooperative fusion of gm-phd filters for online multiple human tracking," *IEEE Trans. Multimedia*, to be published.
[31] Z. Fu, P. Feng, F. Angelini, J. Chambers, and S. M. Naqvi, "Particle PHD filter based multiple human tracking using online group-structured dictionary learning," *IEEE Access*, vol. 6, pp. 14764–14778, 2018.
[32] S. Scheidegger, J. Benjaminsson, E. Rosenberg, A. Krishnan, and K. Granström, "Mono-camera 3D multi-object tracking using deep learning detections and PMBM filtering," in *Proc. IEEE Intell. Vehicles Symp. (IV)*, Changshu, China, Jun. 2018, pp. 433–440.
[33] A. Osep, W. Mehner, M. Mathias, and B. Leibe, "Combined image- and world-space tracking in traffic scenes," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, May/Jun. 2017, pp. 1988–1995.
[34] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *Int. J. Comput. Vis.*, vol. 60, no. 2, pp. 91–110, 2004.
[35] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, Jun. 2005, vol. 1, no. 1, pp. 886–893.
[36] T. Ojala, M. Pietikäinen, and D. Harwood, "A comparative study of texture measures with classification based on featured distributions," *Pattern Recognit.*, vol. 29, no. 1, pp. 51–59, Jan. 1996.
[37] T. Ahonen, A. Hadid, and M. Pietikainen, "Face description with local binary patterns: Application to face recognition," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 28, no. 12, pp. 2037–2041, Dec. 2006.
[38] K. Bernardin and R. Stiefelhagen, "Evaluating multiple object tracking performance: The CLEAR MOT metrics," *J. Image Video Process.*, vol. 2008, pp. 1:1–1:10, Jan. 2008. doi: 10.1155/2008/246309.
[39] Y. Li, C. Huang, and R. Nevatia, "Learning to associate: HybridBoosted multi-target tracker for crowded scene," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2009, pp. 2953–2960.
[40] P. F. Felzenszwalb, R. B. Girshick, D. McAllester, and D. Ramanan, "Object detection with discriminatively trained part-based models," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 32, no. 9, pp. 1627–1645, Sep. 2010.
[41] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards real-time object detection with region proposal networks," in *Proc. Adv. Neural Inf. Process. Syst.*, 2015, pp. 91–99.
[42] F. Yang, W. Choi, and Y. Lin, "Exploit all the layers: Fast and accurate CNN object detector with scale dependent pooling and cascaded rejection classifiers," in *Proc. IEEE Conf. Comput. Vis. pattern Recognit.*, Jun. 2016, pp. 2129–2137.

HASITH KARUNASEKERA received the bachelor's degree in electronics and telecommunication from University of Moratuwa, Sri Lanka, in 2013. He is currently pursuing the Ph.D. degree with the School of Electrical and Electronic Engineering, Nanyang Technological University, Singapore.

Before joining Nanyang Technological University, he was a Telecommunication Engineer for two years, from 2013 to 2015. His research interests include stereo vision, and object detection and tracking.

HAN WANG received the bachelor's degree in computer science from Northeast Heavy Machinery Institute, China, and the Ph.D. degree from the University of Leeds, U.K. He has been an Associate Professor with the School of Electrical and Electronic Engineering, Nanyang Technological University, Singapore, since 1992.

His research interests include computer vision and robotics. He has done significant research work in his research areas and published over 120 top quality international conference and journal papers. He has been invited as a member of Editorial Advisory Board, *The Open Electrical & Electronic Engineering Journal*. He is an Editor of the *Unmanned Systems* journal.

HANDUO ZHANG received the bachelor's degree in automation and the master's degree in pattern recognition and intelligent system from Northeastern University. He is currently pursuing the Ph.D. degree with the School of Electrical and Electronic Engineering, Nanyang Technological University, Singapore, since 2016.

He was an Assistant Researcher with the Shenyang Institute of Automation, Chinese Academy of Sciences, from 2013 to 2015. His research interests include 3D reconstruction in large-scale environment, robot perception, and visual SLAM.