Long-term, Multi-Cue Tracking of Hands in Vehicles

Akshay Rangesh, Eshed Ohn-Bar, and Mohan M. Trivedi

***Abstract*—Hands are a very important cue for understanding and analyzing driver activity, and human activity in general. Vision based hand detection and tracking involve major challenges such as attaining robustness to inconsistencies in lighting and scale, background clutter, object occlusion/disappearance and the large variability in hand shape, size, color and structure. In this paper, we introduce a novel framework suitable for tracking multiple hands online. Assigning tracks to these detections is modeled as a bipartite matching problem with an objective of minimizing the total cost. Both motion and appearance cues are integrated in order to gain robustness to occlusion, fast movement, and interacting hands. Additionally, we study the utility of a left versus right hand classifier to disambiguate hand tracks and reduce ID switches. The proposed tracker shows promise on an extensive, naturalistic, and publicly available driving (VIVA Challenge) dataset [1] by tracking both hands of the driver and the passenger effectively.**

***Index Terms*—Multi-object tracking (MOT), multi-cue integration for tracking, tracking under occlusion, naturalistic driving analysis, advanced driver assistance systems, bipartite matching.**

I. Introduction

With increasing effort and resources being invested on intelligent vehicles, the biggest stumbling block in its widespread use is the guarantee for human safety, both inside the vehicle and outside it. This makes it an absolute necessity for intelligent vehicles to adopt a human centered safety approach by analyzing conditions both inside and outside the vehicle, extracting safety critical information and informing the driver or taking control in an unobtrusive manner to avoid mishaps in the near future. The utility of such a system is summarized by a comprehensive survey on automotive collisions, that demonstrated that a driver was 31% [2] less likely to cause an injury-related collision when he had one or more passengers who could alert him to unseen hazards. Consequently, there is great potential for driver-assistance systems that act as virtual passengers, alerting the driver to potential dangers through aural or visual cues. To design such a system in a manner that is neither distracting nor bothersome, these systems must act like real passengers, alerting the driver only in situations where he appears to be unaware of the possible hazard. This requires a context-aware system that simultaneously monitors the environment and actively interprets the behavior of the driver. By fusing information from inside and outside the vehicle, automotive systems can better model the circumstances that motivate driver behavior.

This paper deals with machine vision approaches for tracking hands in video data captured in naturalistic driving conditions. Inferring information from hand activity is especially important in operated vehicles because it may provide vital information about the state of attentiveness of the driver. Secondary tasks in the vehicle, in particular activities involving drivers hands in the car, were shown to affect certain attention markers such as total eyes off the road [3]. Because driver distraction is a leading cause of car accidents [4], studying where the hands are and what they do in the vehicle has never been a more pressing matter.

Hand tracking may also be considered as a primary task whose output is used to extract higher level semantics. Examples of such applications in the intelligent vehicles domain include gesture recognition [5], [6] for hands-free infotainment and navigation control, analysis of driver hand motion patterns [7], [8], [9] to study preparatory movements for maneuvers and driver distraction during crash and near-crash events. As illustrated, tracking of hands is of considerable use in applications ranging from human machine interaction to active safety systems. In addition to tracking hands, we also assign labels to each track to differentiate the hands of the driver from that of the passenger. This information could be used to influence the behavior of an automated system depending on whether a certain action is performed by the driver or the passenger.

The main contributions of this paper are as follows: We introduce the problem of tracking multiple hands in a naturalistic driving setting and list some common challenges and pitfalls. Motivated by state of the art trackers for single target tracking, we propose a combined tracking-detection framework to provide short yet reliable tracklets, while data association is carried out using a bipartite matching algorithm. We also incorporate a hand type classifier for improving overall tracking performance. Finally, we compare three variants of our proposed algorithm with a baseline multi-target tracking algorithm and note a considerable gain in performance. This validates the treatment of hand tracking as a separate and special case of MOT.

The remainder of this paper is organized as follows: Section 2 briefly enlists the related work in the field. Section 3 gives an in depth description of the proposed method. Section 4 describes the experimental setup, documents the results and makes inferences. Section 5 provides concluding remarks.

II. Related Studies

Hand tracking involves estimating the hand motion using frame-to-frame correspondence of the segmented hand regions or features. All the existing hand trackers typically presume that the hand is visible throughout the sequence.

The 3-D-model-based methods [10] for hand tracking can acquire in-depth and accurate motion data and are capable of coping with occlusions. However, these methods usually

The authors are with the Laboratory for Intelligent and Safe Automobiles (LISA), University of California San Diego, La Jolla, CA 92093-0434, USA. (e-mail: {arangesh, eohnbar, mtrivedi}@ucsd.edu)

TABLE I: Overview of selected studies on hand tracking using monocular color cameras.

Research Study	Camera Perspective	Multi-target Tracking	Algorithmic Approach	Experimental Settings
B Stenger *et al.* (2001) [10]	Frontal, Close up	No	3D model with 27 DOF, Unscented Kalman Filter	Indoors, uncluttered background
Kölsch *et al.* (2004) [11]	Point of view	No	Flocks of Features, KLT tracker	Indoors and Outdoors, unconstrained background and lighting
C Shan *et al.* (2004) [12]	Frontal, Medium shot	No	Mean Shift embedded Particle Filter	Indoors, cluttered background
K Imagawa *et al.* (1998) [13]	Frontal, Medium shot	No	Skin color segmentation, blob generation, Kalman filter	Indoors, uncluttered background
A Argyros *et al.* (2004) [14]	Frontal, Medium shot	Yes	Skin color segmentation, blob generation, object hypothesis tracking	Indoors, uncluttered background
This method	Over the right shoulder, looking forward	Yes	ACF detector, Median Flow tracker, SVM classifier/Bipartite matching	Naturalistic driving

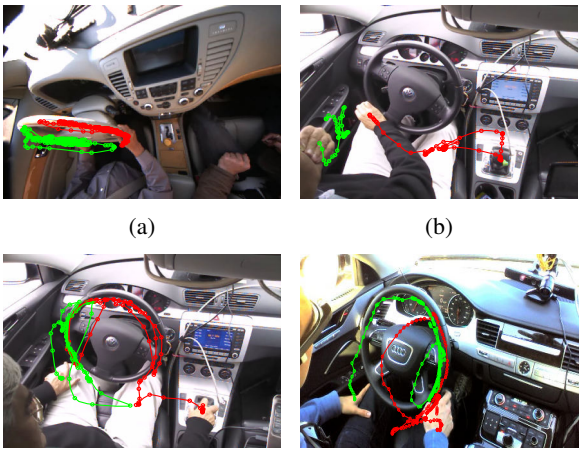
TABLE II: Overview of selected studies on multi-target(object) tracking.

Research Study	Tracker type	Dataset	Target/Object class	Algorithmic Approach	Experimental Settings
L Zhang *et al.* (2008) [15]	Offline	CAVIAR, ETHMS	Pedestrian	Network Flow, Min-cost flow solution	Outdoors, crowded scene
H Pirsiavash *et al.* (2011) [16]	Offline	ETHMS, Caltech Pedestrian	Pedestrian	Min-cost flow network problem, dynamic programming based greedy solution	Outdoors, crowded scene
B Yang *et al.* (2012) [17]	Offline	TUD, Trecvid	Pedestrian	Online learned CRF model, energy postulation & minimization	Outdoors, crowded scene
A Andriyenko *et al.* (2011) [18]	Offline	TUD, PETS etc	Pedestrian & Car	Continuous energy function, gradient descent based optimization	Outdoors, crowded scene
JH Yoon *et al.* (2015) [19]	Online	ETH, YouTube, TUD, PETL1	Pedestrian & Car	Relative Motion Network (RMN), Bayesian filter	Outdoors, crowded scene, moving camera
W Choi *et al.* (2015) [20]	Near online	KITTI, MOT	Pedestrian & Car	Aggregated Local Flow Descriptor (ALFD), NOMT based data association	Outdoors, crowded scene, moving camera

require a complex and expensive hardware setup, suffer from high computational cost and require dense representations of hand articulations.

Blob-based approaches [14], [21] detect hands as image blobs in each frame and temporally correspond blobs that occur in proximate locations across frames. Kalman filter [22] has been employed in works like [23] to transform observations (feature detection) into estimations (extracted trajectory). The advantages are real-time performance, treatment of uncertainty, and the provision of predictions for the successive frames. Certain approaches [24], [25] integrate multiple cues for robust hand tracking. In [24], the authors used color and shape features, along with a combination of particle filter and hidden Markov models (HMM). Table I lists some of the more popular and unique hand tracking algorithms with details of their implementation.

Though hand tracking has been studied extensively in literature, very little effort has been devoted to tracking multiple hand instances simultaneously. MOT algorithms have found considerable success in tracking cars and pedestrians [18], [16] in recent literature. Most of these algorithms use a tracking by detection framework and assign a track to each detection based on solving an optimality criterion. We list some of the most successful algorithms in Table II. Studying these approaches gives us certain insights that we may borrow while solving the multiple hands tracking problem. On the other, it also highlights key differences between generic multi-target tracking and tracking of hands. For instance, it is common for these algorithms to stitch small tracklets (set of overlapping detections) to produce ”smooth” global tracks. However, this



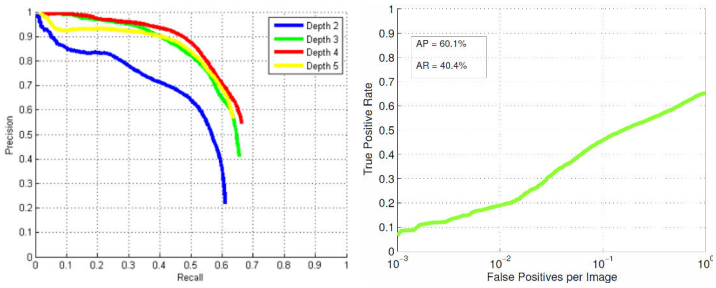
(a) (b) (c) (d)

Fig. 1: Ground truth trajectories of drivers' hands for sample video sequences. Left and right hand tracks are shown green and red respectively.

assumption is invalid in our situation where hand dynamics are highly erratic and almost never smooth e.g. a drivers' hand alternating rapidly between the wheel and gear stick (see Figure 1). Moreover, most of these algorithms are modeled as data association problems that require tracklets generated beforehand. These are called *bacth* or *offline* methods. It is difficult to apply such batch methods to time-critical applications such as ours where safety is of the essence. On the other hand, *sequential* or *online* methods like [26], [19], [27] attempt to resolve ambiguities in each frame (or in a small time window). However, considering more frames before making association decisions should generally help better overcome ambiguities caused by longer-term occlusions and false or missed detections. In this study, we propose a sequential (online) method that leverages information obtained from multiple modalities to ensure correct tracks and their corresponding labels are made available as soon as possible.

III. Proposed Multi-Cue Tracking Framework

As pointed out in [28], the long-term tracking problem is generally approached either from tracking or from detection perspectives. Trackers generally use information from temporally adjacent frames to relocate objects in the current frame. However, trackers are prone to drift and fail when the object is either occluded or completely leaves the frame. Detection-based algorithms estimate the object location in every frame independently. Detectors experience no drift and can detect objects that re-enter the frame. On the other hand, they may produce many false positives or fail to detect true negatives leading to an improper assignment of tracks. Hence, we propose to integrate the tracker and detector in a mutually beneficial manner to overcome each others' individual shortcomings. Figure 3 shows the block diagram of the proposed algorithm. The following subsections are devoted to explaining each individual block in detail.



(a) PR curves using boosted trees of depth 2, 3, 4, and 5.

(b) Final ROC curve with tuned parameters.

Fig. 2: Performance metric for hand detector.

A. Hand Detection

We use an ACF detector [29] to provide bounding box estimates in each frame. This approach is carefully tuned for hand detection in the vehicle, and it incorporates both color and edge cues which are vital for hand detection. The detector was trained using 10 channels (LUV + Normalized Gradient Magnitude + 6 x Gradient Orientation). The detector cascade consists of 4 stages of AdaBoost with 32, 128, 512, and 2048 weak learners for each stage respectively. The weak learner chosen was a depth 4 decision tree. A depth greater than 4 resulted in over-fitting (Figure 2(a)). The template height was set at 65 pixels with an aspect ratio of 0.9. Further details about the training procedure, choice of parameters, and performance of the ACF detector can be found in [30]. The choice of an ACF detector ensures detection at multiple scales while maintaining a relatively low computational cost. Figure 2(b) shows the ROC curve for the ACF detector on the training dataset. **AP** denotes the percentage area under the precision recall curve. **AR** (average recall rate) is calculated over 9 evenly sampled points in log space between 10^{-2} and 10^{0} false positives per image. As can be inferred, objects with high degree of freedom like the human hand are difficult to detect because of their large state space, complex image appearance and high variability in pose. We try to overcome the inherent limitations of a hand detector by using a tracker in conjunction.

B. Median Flow Tracking

The hand detector alone is found to fall short of producing smooth object trajectories in challenging naturalistic driving settings. The detector fails to detect some hand instances, while also introducing occasional false positives. We propose the use of a modified median flow tracker [31] to solve these issues.

Given an set of ground truth hand bounding boxes, we initialize a set of keypoints [32] within each box to track in the next frame. The sparse motion flow of these keypoints are then determined using the pyramidal Lucas-Kanade algorithm [33]. Only points with a bidirectional (Forward-Backward) error less than 3 pixels are retained for the voting step. Additionally, points with a low skin likelihood are discarded to ensure the final track location remains on the hand. This also prevents keypoints that are not localized on the hand from dominating

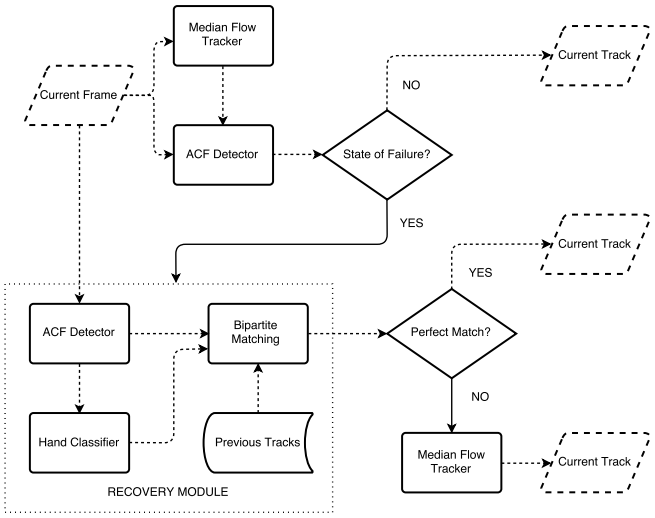


Fig. 3: Block diagram of proposed tracking framework for a given time-step. Solid arrows indicate control transfer, while dashed arrows indicate data transfer.

the voting procedure, thereby reducing the drift considerably. The median along both spatial dimensions (in the image plane) of all retained keypoints gives us the track location of the object for the given frame.

The median flow tracker may also be exploited to provide a bounding box estimate based on scale change. Scale change is computed as follows: for each pair of points, a ratio between current point distance and previous point distance is computed; bounding box scale change is defined as the median over these ratios. An implicit assumption of the point-based representation is that the object is composed of small rigid patches. Parts of the objects that do not satisfy this assumption (object boundary, flexible parts) are not considered in the voting since they are rejected by the error measure. The bounding box is centered about the current track location.

C. Integration

Given the track location for each object in a given frame, the ACF detector is run independently on a region of interest (*RoI*) around each track location. The size of the *RoI* is chosen to be twice that of the corresponding box in the previous frame, centered about the track location in the current frame. The advantage of this is two fold; We eliminate almost all false positives of the detector by limiting the search area to a few chosen regions. Also, by re-initializing keypoints within the final bounding box predicted by the detector, we reset any drift that seeps into the tracker. This ensures a reliable track for the next frame and eventually a reliable detection. This symbiotic relationship between the tracker and detector is found to keep up with fast and complex hand movements that occur regularly while driving.

In most cases, the sliding window search returns one high scoring bounding box per hand. If multiple boxes are detected, the bounding box with the highest overall score is chosen. Let t denote the current frame that is under process and $i = 1, 2, \cdots, N$ denote a particular bounding box returned by the detector for any given object $j = 1, 2, \cdots, M$. The overall score of a bounding box $S^t_{i,j}, i = 1, 2, \cdots, N$ is then assigned as follows:

$$S^t_{i,j} = f^t_{i,j} \cdot exp(-\lambda|(h^{t-1}_{m,j} - h^t_{i,j})(w^{t-1}_{m,j} - w^t_{i,j})|), \quad (1)$$

where $f^t_{i,j}$ denotes the fraction of keypoints belonging to object j enclosed by bounding box i for a given frame t and $h^t_{i,j}$, $w^t_{i,j}$ denote the height and width respectively of the corresponding bounding box. Also, $h^{t-1}_{m,j}, w^{t-1}_{m,j}$ denote the best scoring bounding box of frame $t-1$, i.e. the previous frame. The constant λ is determined experimentally. The first term in the above scoring mechanism ensures that the detector output fits the trajectory estimated by the tracker, thereby reinforcing our belief in the overall output. The second term acts as a reguralizer, preventing the selection of larger bounding boxes with relatively poor fits.

The final bounding box for an object j in frame t is given as:

$$O^t_j = O^t_{m,j}, \quad (2)$$

where

$$m = \underset{i}{\operatorname{argmax}}\, S^t_{i,j}, \forall i \in 1, 2, \cdots, N. \quad (3)$$

On the off chance that there is no high scoring bounding box proposal, or no proposal at all, the one obtained from the median flow tracker is used. This keeps a track on the object until the detector is functional again. This proves to be essential in keeping up with variation in hand pose or harsh lighting conditions which renders the detector temporarily unreliable.

D. Failure Detection and Recovery

The combined tracker-detector framework inevitably fails during long video sequences set in a naturalistic driving setting. We document the most common reasons for failure below:

- Two separate hand tracks merge into one when in proximity of one another (Figure 4(a-d)). This is caused by overlapping regions of interest.
- Temporary self occlusion of a hand instance leading to ID switches (Figure 4(e-h)).
- Tracked object leaves the frame and re-enters at a future instance (Figure 4(i-l)).

Our intent is to detect such failures and reinitialize the tracks appropriately to ensure long-term tracking. To do so, we declare the system to be in a *state of failure* when one or more of the following occurs:

- If two different hand tracks are associated with the same (or similar) bounding box proposal. This condition covers the second problem mentioned above.

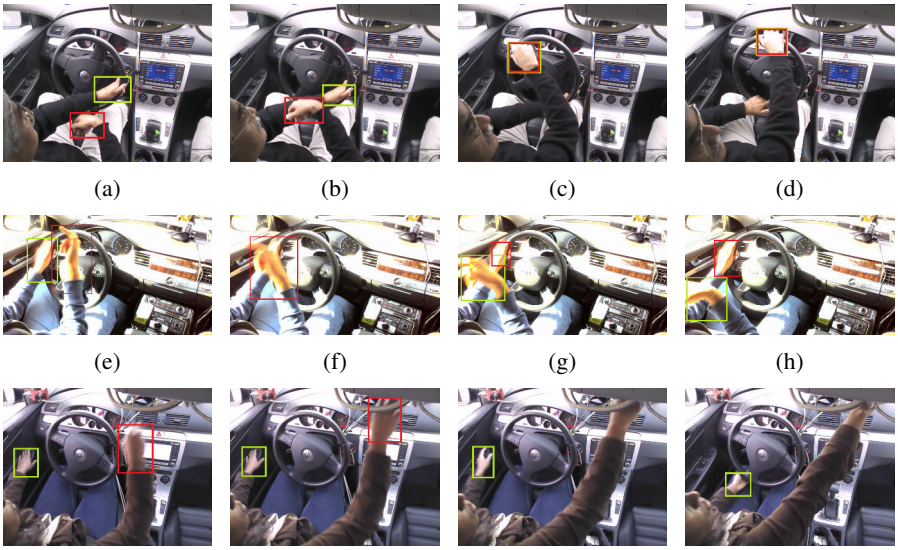


Fig. 4: Examples that cause a *state of failure*. Each detection is color coded to denote separate tracks.

- If the ACF detector is unable to provide a high scoring tracklet for 10 continuous frames. This condition informs us when the object has left the frame, is occluded or when the track has completely drifted away from the ground truth object location.

Once in a *state of failure*, the control is handed over to the detector, which sweeps through the whole image and outputs bounding box proposals. Each bounding box is assigned a weight (or cost) based on its proximity to the last known location of each track. The activity of optimally assigning each of N tracks, one out of M bounding box proposals each is modeled as a weighted bipartite matching problem. We use the popular Hungarian algorithm to obtain a perfect match. If a perfect or finite cost match is not possible in the current frame, the system continues to stay in a *state of failure* with updated tracks.

E. Hand Type Classification

In addition to finding a perfect match as mentioned above, we also studied the effectiveness of a left versus right hand classifier. The utility of such a classifier is obvious for many reasons. One obvious advantage would be to disambiguate track labels thereby reducing ID switches. It would also reduce the large number of false matches observed when two hand tracks cross over each other, by assigning correct tracks after they diverge.

To check the feasibility of such a classifier, we extracted left and right hand instances using ground truth annotations from the VIVA training dataset for hand tracking. Since both hand classes would exhibit similar color profiles, we limited our testing to gradient and structure based descriptors i.e. HOG and CNN. The method used to extract each descriptor is given below:

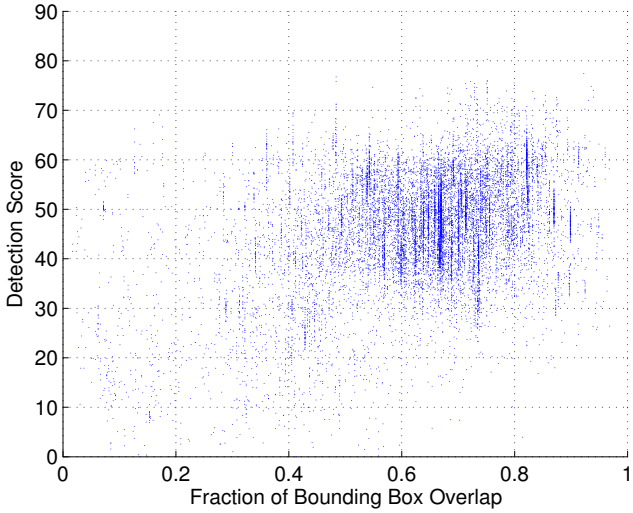


Fig. 5: Scatter plot of detection scores (from ACF detector) versus fraction of bounding box overlap with ground truth. As can be seen, most successful detections ($overlap \geq 0.5$) have relatively high scores (≥ 30).

- **HOG** : The input image patch is re-sized to 48 x 48 and HOG features are extracted for a spatial bin size of 8 pixels and 9 orientation bins. The final descriptor is obtained by vectorizing the HOG features.
- **CNN** : Inspired from [34], we use a generic pre-trained model to extract the CNN descriptor. Since deeper models are seen to generalize well to a wide range of tasks and datasets, we choose the 16 layer very deep ConvNet configuration proposed in [35]. Once each layer of the

network is evaluated, we simply use the L2 normalized 4096 length vector in the penultimate (FC) layer as the descriptor.

In both cases, an SVM classier [36] with a linear kernel was evaluated using one fold validation (training/validation split). The linear kernel SVM has a single parameter C (penalty parameter) which is optimally chosen using an exhaustive search in parameter space. It is found that the parameter C offers a small improvement in performance beyond a value of 2. We use this value throughout our experiments to prevent over-fitting to the training data. The training/validation split was carried out in four different ways:

- **Type 1**: The set of ground truth hand regions were randomly partitioned into a training set and validation set respectively.
- **Type 2**: The set of video sequences were partitioned into training and validation sets in a manner that ensures cross-subject, cross-drive validation. The ground truth hand regions extracted from these two sets are used for training and validation respectively.
- **Type 3**: The split is done in the same way as Type 2. The only difference here is that instead of using ground truth hand regions, we use the bounding box proposals from the ACF detector which have an overlap of greater than or equal to 50% with the ground truth.
- **Type 4**: The split is done the same way as Type 2, except we extend the ground truth bounding boxes by 10% in all directions.

TABLE III: Accuracy of left/right hand classifier during validation.

Validation Method	HOG	CNN
Type 1	99.5359%	99.0853%
Type 2	84.0530%	94.1987%
Type 3	86.1600%	95.0600%
Type 4	89.9936%	95.1027%

Table 3 lists the classification accuracies obtained for each validation strategy and for both descriptors respectively. As can be seen, *Type 1* validation indicates near perfect accuracy. This may be attributed to the fact that both training and validation sets contain temporally adjacent hand instances. Hence, the classification is much easier as the classifier is both trained and validated on very similar data. This motivates the split used in *Type 2*. As expected, the accuracies are lower in this case. For the first two types, the classifier was evaluated on tightly fit ground truth bounding boxes. This however, will not be the case when we integrate the classifier into our tracking framework. To test it on harder examples, *Type 3* validation was chosen, which would mirror the conditions that the classifier would face when integrated into the tracking framework. Surprisingly, the classifier accuracies are slightly higher than those from *Type 2* validation. This indicates that for tightly fit bounding boxes (or ground truth), the descriptors fail to capture information from regions surrounding the hand, which causes a slight performance degrade. This hypothesis is validated using the *Type 4* split, which shows an improvement in performance over the *Type 2* split.

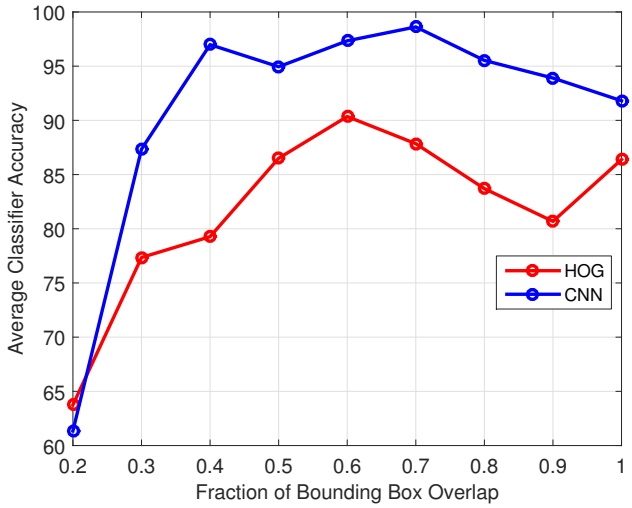


Fig. 6: Plot for average classifier accuracy versus fraction of bounding box overlap (with ground truth).

To see how the accuracy changes with the bounding box overlap, we again split the video sequences into three independent sets and perform three fold cross validation. This time, we train the classifier on all detector proposals with an overlap greater than 20% and test it on subsets of data from the ACF detector, split based on their overlap fraction with the ground truth (from 0.2 to 1 in steps of 0.1). Figure 6 shows the plot for the same. The CNN based descriptor seems to outperform the HOG descriptor for almost all fractions of overlap. Surprisingly, the HOG descriptor performs best for overlap fractions between 0.6 and 0.7, and slowly degrades as the overlap increases. This may be because a tight fitted region proposal (i.e. high overlap fraction) does not provide enough gradient information around the hand. However, the CNN descriptor performs consistently well for any overlap greater than and equal to 0.4. For low overlap fractions, both descriptors perform poorly considering the lack of training data.

The final SVM classifier is trained on the union of all ground truth hand regions and all detector proposals with an overlap of greater than 20% with the ground truth. Additionally, we augment the training data by synthetically generating bounding boxes with small overlap ($\leq 40\%$) to ensure the classifier is able to handle loosely fit proposals as well. This makes the classifier output more robust to any localization errors and jitters produced by the ACF detector.

F. Bipartite Matching for Data Association

Consider the problem of matching each of N tracks to one of M bounding box proposals. We formulate the probability of associating a track $T_i, i = 1, 2, \cdots, N$ with an object (bounding box) $O_j, j = 1, 2, \cdots, M$ as the product of three components (distance, classifier, detector):

$$P(O_j \in T_i) = P_{dist}(O_j \in T_i) P_{class}(O_j \in T_i) P_{det}(O_j \in T_i) \tag{4}$$

Distance P_{dist} encodes the distance between the last know track location and the current object (bounding box) location:

$$P_{dist}(O_j \in T_i) = \frac{d_{i,j^*}}{d_{i,j}}, \tag{5}$$

where

$$j^* = \underset{j}{\operatorname{argmin}}\, d_{i,j}, \ \forall j = 1, 2, .., M. \tag{6}$$

$d_{i,j}$ denotes the distance between the latest location of track T_i and the center of the object O_j.

Classification $P_{class}(O_j \in T_i)$ gives the probability that an object O_j has the same class (left/right hand) as that of the track T_i. This probability is obtained from the SVM classifier.

Detection P_{det} ensures that false positives are weeded out before data association. It is defined as follows:

$$P_{det}(O_j \in T_i) = \begin{cases} 1 & \text{if score}(O_j) \geq 30, \\ 0 & \text{otherwise}, \end{cases} \tag{7}$$

where the score is obtained from the ACF detector. As can be inferred from Figure 5, most successful detections (overlap ≥ 0.5) have a score higher than 30.

Integrating three different cues while assigning probabilities helps handle the data association problem effectively by taking a more holistic view of the situation, thereby pushing the matching algorithm to reason over distance in consecutive frames, class, and detection score confidence. This allows regularization when resolving difficult cases, for example, when two hands interact or overlap one another.

To find the optimal assignment between tracks and objects, we need to form an NxM cost matrix $C = \{C_{i,j}\}$, with

$$C_{i,j} = -\log P(O_j \in T_i) \tag{8}$$

and then apply Hungarian algorithm to find the min-cost solution. If a finite cost solution is not found, the system remains in a *state of failure* with updated tracks from median flow tracker. This conservative approach ensures reliability of tracks in the long term.

IV. Experimental Evaluations

The VIVA challenge [1] dataset for hand tracking is used to evaluate each tracker. The dataset consists of 27 hand annotated training sequences and 29 testing sequences captured under naturalistic driving conditions with both driver and passenger in the field of view. The dataset was captured under real world conditions and offers challenges such as different subjects, different cars, different capture settings, different perspectives, background clutter and harsh illumination. We test three variants of the proposed tracker, the details of which are given below:

- **TD** : Combined tracking and detection framework without hand type classifier.
- **TDC**$_{HOG}$: Combined tracking and detection framework with HOG based hand type classifier.
- **TDC**$_{CNN}$: Combined tracking and detection framework with CNN based hand type classifier.

To evaluate the performance the proposed framework, we also provide the results of a baseline multi-target tracking by detection algorithm proposed in [37]. This tracker operates in three stages: First, objects are detected in each frame independently using the DPM object detector. Second, all detections with a positive score are associated to detections in the next frame using appearance and the bounding box overlap. Prediction is performed using a Kalman filter and detections are associated between both frames via the Hungarian method for bipartite matching. To gap occlusions and missed detections, tracklets are associated with each other in a third stage. Similarly to the second stage, the Hungarian algorithm is employed but this time based on an occlusion sensitive appearance model and regression of the bounding boxes in one tracklet from the bounding boxes in the other tracklet. The algorithm outputs all associated tracklets which are longer than three frames. To ensure a fair comparison, all trackers use the same detections provided by the ACF detector.

The following metric [38], [39] are used to evaluate the performance of the tracker:

- **The multiple object tracking accuracy (MOTA)**:

$$MOTA = 1 - \frac{\sum_t (m_t + fp_t + mme_t)}{\sum_t g_t}, \tag{9}$$

  where m_t, fp_t, mme_t and g_t are the number of misses, false positives, mismatches, and number of ground truth objects respectively, for time t.
  The MOTA accounts for all object configuration errors made by the tracker, false positives, misses, mismatches, over all frames. It gives a very intuitive measure of the trackers performance at detecting objects and keeping their trajectories, independent of the precision with which the object locations are estimated.
  The MOTA has an upper bound of 1, and is unbounded from below. It can assume negative values in extreme cases.

- **The multiple object tracking precision (MOTP)**:

$$y_1 = \frac{\sum_{i,t} d_t^i}{\sum_t c_t}, \tag{10}$$

  where c_t is the number of matches for time t and d_t^i is the distance between object o_i and its corresponding hypothesis.
  It is the total error in estimated position for matched object-hypothesis pairs over all frames, averaged by the

total number of matches made. It shows the ability of the tracker to estimate precise object positions, independent of its skill at recognizing object configurations, keeping consistent trajectories, and so forth.
The MOTP has a lower bound of 0 (corresponding to exact localization), and is unbounded from above.

- **Mostly Tracked (MT)** : Fraction of GT trajectories which are covered by tracker output for more than 80% in length. MT assumes a value in [0,1].
- **Mostly Lost (ML)** : Fraction of GT trajectories which are covered by tracker output for less than 20% in length. ML assumes a value in [0,1].
- **ID Switches (IDS)**: The total of number of times that a tracked trajectory changes its matched GT identity. IDS takes on non-negative integral values, with an IDS of 0 corresponding to the ideal case (no ID switches).
- **Fragments (Frag)**: The total of number of times that a GT trajectory is interrupted in tracking result. Frag takes on non-negative integral values, with a Frag equal to 0 corresponding to the ideal case (no Fragmentation).

The reason for using more than one metric for evaluation is simply because no one metric alone is representative of the overall performance of a multi-object tracker. With this in mind, a tracker would be considered to be good enough only if all its metric are reasonably good. For example, having a low MOTP (small localization error) would not matter if the MOTA is too low. The evaluation metric for the whole test dataset are listed below:

TABLE IV: Results on VIVA Challenge dataset.
Arrows (↑ / ↓) corresponding to each metric indicate if a high or low value is desired.

Method	Type	MOTA (↑)	MOTP (↓)	MT (↑)	ML (↓)	IDS (↓)	Frag (↓)
TD	Online	0.215342	0.645499	0.359375	0.171875	46	418
TDC$_{HOG}$	Online	0.246675	**0.645438**	0.361244	0.174359	39	426
TDC$_{CNN}$	Online	**0.250920**	0.645655	0.390625	0.187500	37	415
Baseline	Offline	0.067541	0.659595	**0.500000**	**0.125000**	**29**	**320**

As can be seen above, all three versions of the proposed tracker vastly improve upon the baseline in terms of overall accuracy. The huge gain in **MOTA** may be attributed to reduced false positives and missed detections. This suggests that combining tracking and detection in a mutually beneficial manner is superior to the tracking-by-detection framework for our purpose (Figure 8). The **MOTP** in this case is not too informative considering all trackers use the same detection boxes. This is reflected in the MOTP scores given above.

It is also seen that integrating a hand classifier does improve the overall performance, albeit not by a huge margin. This is misleading because all three versions essentially operate in the same manner unless in a *state of failure*. Since a *state of failure* occurs sporadically, the improvement in performance metric over the whole dataset may seem insignificant.

Both, the CNN and HOG descriptor based classifier reduce **IDS** in hard situations despite the fact that it is processed online (Figure 7). This suggests that training a left/right hand classifier offers a unique advantage that may be used for tracking hands.

The Baseline tracker (offline) shows superior **Frag**, **IDS** and **MT**. This is natural because our method is an online method which does not use any future information. Therefore, some short tracks are not fused together leading to a higher Frag and a lower MT.

The combined tracking and detection framework without the hand classifier runs at 12fps for a 640 by 480 video on a modern CPU. When the hand classifier is integrated, the time taken by the Hungarian algorithm (of order $O(n^4)$) is relatively small as $n_{max} = 4$. Also, the SVM classifier with a linear kernel is computationally cheap. The primary bottleneck in speed comes from feature extraction. However, this does not affect the overall performance as the operation is needed only in a *state of failure* (which occurs once every 150 frames on average). This results in the entire framework with the hand classifier running at 11 and 9fps (on average) for the HOG and CNN descriptor respectively.

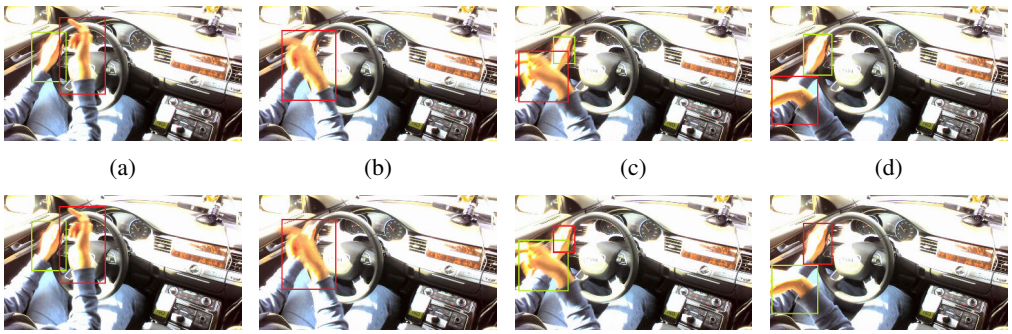
V. Concluding Remarks

This paper introduces a novel multi-cue tracking framework designed for long-term analysis of vehicle occupants' hand movements. The uniqueness and benefits of such a tracker in comparison to a generic MOT are highlighted, and a case is made for its separate consideration. A combined tracking and detection framework is proposed to produce individual tracks online, and data association is performed using the Hungarian algorithm. Although motion and appearance-based tracking is clearly motivated, these alone provide difficult disambiguation when the hands are occluded or interacting. Therefore, a hand type classifier was integrated and shown to greatly reduce IDS, which validates its utility. The proposed algorithm is shown to significantly improve over a state of the art baseline, despite its online operation.

Further improvements may be obtained by extracting improved motion features [40], texture features [41] and using a more robust hand detector by learning multiple models [42]. Extracting high level semantics from hand tracklets is left for future work.

VI. Acknowledgments

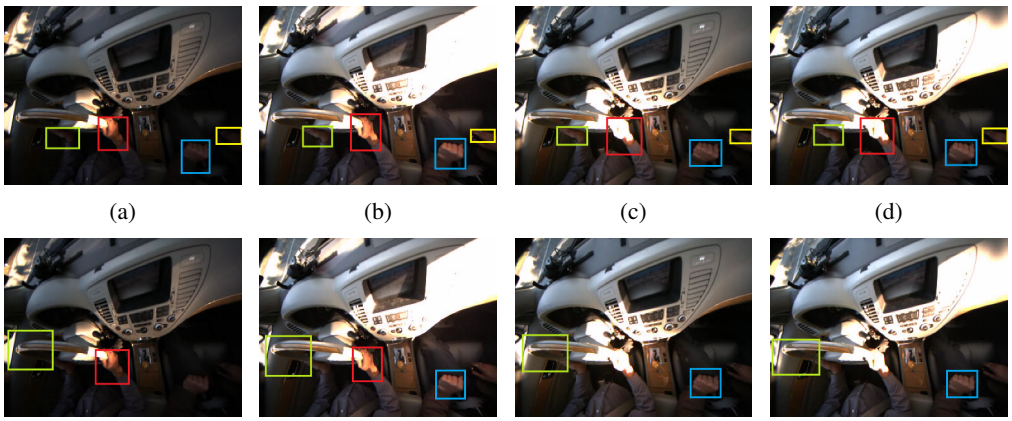
The authors would like to thank the reviewers and the editors for their constructive suggestions and comments. We also acknowledge the support of our sponsors. Last but not least, we would like to thank our colleagues from the Laboratory for Intelligent and Safe Automobiles (LISA), UCSD for their support, and especially Rakesh N. Rajaram for his contribution in the evaluation phase of our research.



(a) (b) (c) (d)

(e) (f) (g) (h)

Fig. 7: Example results during a possible ID switch. Here, the top row (a-d) depicts the tracking result of **TDC**$_{CNN}$ and **Baseline** while the bottom row (e-f) show that of **TD** and **TDC**$_{HOG}$. Individual tracks are color coded for convenience. As can be seen, **TDC**$_{CNN}$ and the **Baseline** prevent the ID switch, whereas the other two cannot.



(a) (b) (c) (d)

(e) (f) (g) (h)

Fig. 8: Example results during when the detector fails. Here, the top row (a-d) depicts the tracking result of **TDC**$_{HOG}$, **TDC**$_{CNN}$ and **TD** while the bottom row (e-f) show that of the **Baseline**. Individual tracks are color coded for convenience. In this case, the **Baseline** is unable to deal with missed detections and false positives generated by the detector. The proposed framework handles this by relying on the tracker until the detector becomes functional.

REFERENCES

[1] Computer Vision and Robotics Research Laboratory, UCSD, "Vision for Intelligent Vehicles and Applications (VIVA)," http://cvrr.ucsd.edu/vivachallenge/.

[2] R. L. Olson, R. J. Hanowski, J. S. Hickman, and J. L. Bocanegra, "Driver distraction in commercial vehicle operations," Tech. Rep., 2009.

[3] S. G. Klauer, F. Guo, J. Sudweeks, and T. A. Dingus, "An analysis of driver inattention using a case-crossover approach on 100-car data:final report," Tech. Rep., 2010.

[4] J. Tison, N. Chaudhary, and L. Cosgrove, "National phone survey on distracted driving attitudes and behaviors," Tech. Rep., 2011.

[5] E. Ohn-Bar and M. M. Trivedi, "Hand gesture recognition in real time for automotive interfaces: A multimodal vision-based approach and evaluations," *IEEE Transactions on Intelligent Transportation Systems*, 2014.

[6] E. Ohn-Bar and M. M. Trivedi, "The power is in your hands: 3d analysis of hand gestures in naturalistic video," in *IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, 2013.

[7] E. Ohn-Bar and M. M. Trivedi, "Beyond just keeping hands on the wheel: Towards visual interpretation of driver hand motion patterns," in *IEEE Conference on Intelligent Transportation Systems*, 2014.

[8] C. Tran and M. M. Trivedi, "3-d posture and gesture recognition for interactivity in smart spaces," *IEEE Transactions on Industrial Informatics*, 2012.

[9] E. Ohn-Bar, A. Tawari, S. Martin, and M. M. Trivedi, "On surveillance for safety critical events: In-vehicle video networks for predictive driver assistance systems," *Computer Vision and Image Understanding*, 2015.

[10] B. Stenger, P. R. Mendonça, and R. Cipolla, "Model-based 3d tracking of an articulated hand," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2001.

[11] M. Kölsch and M. Turk, "Fast 2d hand tracking with flocks of features and multi-cue integration," in *IEEE Conference on Computer Vision and Pattern Recognition Workshop*, 2004.

[12] C. Shan, Y. Wei, T. Tan, and F. Ojardias, "Real time hand tracking by combining particle filtering and mean shift," in *Sixth IEEE International Conference on Automatic Face and Gesture Recognition*, 2004.

[13] K. Imagawa, S. Lu, and S. Igi, "Color-based hands tracking system for

sign language recognition," in *Third IEEE International Conference on Automatic Face and Gesture Recognition*, 1998.

[14] A. A. Argyros and M. I. Lourakis, "Real-time tracking of multiple skin-colored objects with a possibly moving camera," in *Computer Vision-ECCV 2004*.

[15] L. Zhang, Y. Li, and R. Nevatia, "Global data association for multi-object tracking using network flows," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2008.

[16] H. Pirsiavash, D. Ramanan, and C. C. Fowlkes, "Globally-optimal greedy algorithms for tracking a variable number of objects," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2011.

[17] B. Yang and R. Nevatia, "An online learned crf model for multi-target tracking," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2012.

[18] A. Andriyenko and K. Schindler, "Multi-target tracking by continuous energy minimization," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2011.

[19] J. H. Yoon, M.-H. Yang, J. Lim, and K.-J. Yoon, "Bayesian multi-object tracking using motion context from multiple objects," in *IEEE Winter Conference on Applications of Computer Vision (WACV)*, 2015.

[20] W. Choi, "Near-online multi-target tracking with aggregated local flow descriptor," *arXiv:1504.02340*, 2015.

[21] H. Birk, T. B. Moeslund, and C. B. Madsen, "Real-time recognition of hand alphabet gestures using principal component analysis," in *Proceedings of the Scandinavian Conference on Image Analysis*, 1997.

[22] R. E. Kalman, "A new approach to linear filtering and prediction problems," *Journal of Fluids Engineering*, 1960.

[23] M. Breig and M. Kohler, *Motion detection and tracking under constraint of pan tilt cameras for vision based human computer interaction*, 1998.

[24] H. Fei and I. Reid, "Probabilistic tracking and recognition of nonrigid hand motion," in *IEEE International Workshop on Analysis and Modeling of Faces and Gestures*, 2003.

[25] Q. Yuan, S. Sclaroff, and V. Athitsos, "Automatic 2d hand tracking in video sequences," in *Seventh IEEE Workshops on applications of Computer Vision*, 2005.

[26] Z. Khan, T. Balch, and F. Dellaert, "Mcmc-based particle filtering for tracking a variable number of interacting targets," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2005.

[27] A. Gaidon and E. Vig, "Online domain adaptation for multi-object tracking," *arXiv:1508.00776*, 2015.

[28] Z. Kalal, K. Mikolajczyk, and J. Matas, "Tracking-learning-detection," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2012.

[29] P. Dollár, "Piotr's Computer Vision Matlab Toolbox (PMT)," http://vision.ucsd.edu/~pdollar/toolbox/doc/index.html.

[30] E. Ohn-Bar, N. Das, and M. M. Trivedi, "On performance evaluation of driver hand detection algorithms: Challenges, dataset, and metrics," in *IEEE Conference on Intelligent Transportation Systems*, 2015.

[31] Z. Kalal, K. Mikolajczyk, and J. Matas, "Forward-backward error: Automatic detection of tracking failures," in *20th International Conference on Pattern Recognition (ICPR)*, 2010.

[32] J. Shi and C. Tomasi, "Good features to track," in *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 1994.

[33] B. D. Lucas, T. Kanade *et al.*, "An iterative image registration technique with an application to stereo vision." in *IJCAI*, 1981.

[34] A. S. Razavian, H. Azizpour, J. Sullivan, and S. Carlsson, "Cnn features off-the-shelf: an astounding baseline for recognition," in *IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, 2014.

[35] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *arXiv:1409.1556*, 2014.

[36] C.-C. Chang and C.-J. Lin, "Libsvm: A library for support vector machines," *ACM Transactions on Intelligent Systems and Technology (TIST)*, 2011.

[37] A. Geiger, M. Lauer, C. Wojek, C. Stiller, and R. Urtasun, "3d traffic scene understanding from movable platforms," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2014.

[38] K. Bernardin and R. Stiefelhagen, "Evaluating multiple object tracking performance: the clear mot metrics," *Journal on Image and Video Processing*, 2008.

[39] Y. Li, C. Huang, and R. Nevatia, "Learning to associate: Hybridboosted multi-target tracker for crowded scene," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2009.

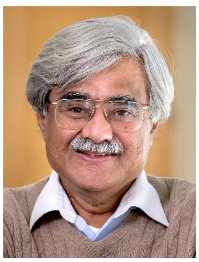
[40] H. Wang, A. Kläser, C. Schmid, and C.-L. Liu, "Dense trajectories and motion boundary descriptors for action recognition," *International journal of computer vision*, 2013.

[41] M. M. Trivedi, C. A. Harlow, R. W. Conners, and S. Goh, "Object detection based on gray level cooccurrence," *Computer Vision, Graphics, and Image Processing*, 1984.

[42] E. Ohn-Bar and M. M. Trivedi, "Learning to detect vehicles by clustering appearance patterns," *IEEE Transactions on Intelligent Transportation Systems*, 2015.

Akshay Rangesh is currently pursuing a Master's degree in electrical engineering from the University of California, San Diego (UCSD) with a focus on Intelligent Systems, Robotics & Control. His research interests span computer vision and machine learning, with a focus on object detection and tracking, human activity recognition and driver safety systems.

Eshed Ohn-Bar received his M.S. degree in electrical engineering from the University of California, San Diego (UCSD) in 2013 and is currently pursuing a Ph.D. with a focus on signal and image processing at UCSD. His research interests include computer vision, object detection, multi-modal activity recognition, intelligent vehicles, and driver assistance and safety systems.



Mohan Manubhai Trivedi is a distinguished Professor of Electrical and Computer Engineering and the founding director of the UCSD LISA: Laboratory for Intelligent and Safe Automobiles, winner of the IEEE ITSS Lead Institution Award (2015). Trivedi is a fellow of IEEE, SPIE and IAPR. Currently, Trivedi and his team are pursuing research in intelligent vehicles, machine perception, machine learning, human-robot interactivity, driver assistance, active safety and embedded systems. He received the IEEE ITS Society's highest honor, the "Outstanding Research Award" in 2013. He serves regularly as a consultant to industry and government agencies in the USA and abroad.