DeepTAgent: Realtime Tracking of Dense Traffic Agents Using Heterogeneous Interaction

Rohan Chandra[1], Tanmay Randhavane[2], Uttaran Bhattacharya[1], Aniket Bera[2], and Dinesh Manocha[1]
http://gamma.cs.unc.edu/HTI

***Abstract*— We present a realtime algorithm to track different traffic agents in dense videos. Our approach is designed for heterogeneous traffic scenarios, which consist of different agents including vehicles, bicycles, pedestrians, two-wheelers, etc., sharing the road. We present a novel heterogeneous traffic motion and interaction model (HTMI) to predict the trajectories and interaction between the agents. We combine HTMI with the tracking-by-detection paradigm and use CNNs to compute the features of traffic agents for accurate tracking reliably. We highlight the performance on a new dataset of dense traffic videos and observe 72.02% accuracy. Our approach can handle all kinds of traffic videos in realtime on a single GPU. We observe $4X$ speedup over prior tracking algorithms and more than 7% improvement in accuracy.**

I. Introduction

Tracking of traffic agents on a highway or an urban road is an important problem in autonomous driving, computer vision, intelligent transportation, and related areas. These *heterogeneous* traffic agents may correspond to large or small vehicles, buses, bicycles, rickshaws, pedestrians, moving carts, etc. Different agents have different shapes, move at varying speeds, and their trajectories are governed by underlying dynamics constraints. Furthermore, the traffic patterns or behaviors can vary considerably between highway traffic, urban traffic or driving in highly congested areas (e.g., in Asian cities).

Given a traffic video, the tracking problem corresponds to computing the consistency in the temporal and spatial identity of all agents in the image sequence. There is extensive work in vehicle and object tracking, and different methods have been proposed based on the use of cameras or laser rangefinders. Recent developments in autonomous driving and large-scale deployment of high-resolution cameras for surveillance has generated interest in the development of realtime tracking algorithms, especially in dense scenarios with a large number of heterogeneous agents. The complexity of tracking increases as different types of traffic agents are in close proximity and interact with each other, including vehicle-vehicle, vehicle-pedestrian, vehicle-bicycle, pedestrian-pedestrian, etc.

The traffic congestion on highways and urban roads often result in high-density traffic scenarios. The traffic density can be defined based on the number of distinct traffic agents captured in a single frame of the video or the number of agents per unit length (e.g., a kilometer) of the roadway. It is not uncommon to capture videos with tens or hundreds of traffic agents in a single frame. The high density makes it hard to track all the agents reliably over a sequence of frames.

Fig. 1: *We highlight the performance of our tracking algorithm, DeepTAgent, in this urban video. This frame consists of 27 traffic-agents, including pedestrians, two-wheel scooters, three-wheel rickshaws, cars, and bicycles. We can compute the trajectories in realtime with 72.02% accuracy.*

Main Contributions: We present a novel realtime algorithm (DeepTAgent) to track traffic agents in dense videos. Our approach makes no assumptions about the agents or their motion or lighting conditions. We combine deep learning based agent detection scheme (i.e., tracking-by-detection paradigm) with a novel heterogeneous traffic motion and interaction (HTMI) model to compute the features of each agent and perform reliable tracking. The novel contributions of our work include:

1) We present "Heterogeneous Traffic Motion and Interaction Model" (*HTMI*), to model the motion of different traffic agents as well as the pairwise interaction between nearby agents. It is general and applicable to all traffic agents.
2) We present an algorithm to automatically compute *Deep TA-Features* of each agent using HTMI and CNNs. These features reduce the number of false negatives and identity switches and significantly increase the accuracy of our algorithm in dense videos.
3) We have a collected a high-resolution traffic dataset corresponding to dense videos of highway and urban traffic with varying lighting conditions and camera angles. Furthermore, these videos capture the traffic patterns and behaviors from different geographic regions of the world, including the USA, China, and India. We

[1]Author from the Department of Computer Science and Electrical & Computer Engineering, University of Maryland at College Park, USA
[2]Authors from the Department of Computer Science, University of North Carolina at Chapel Hill, USA

have evaluated the performance of DeepTAgent and observe 72.02% tracking accuracy.

We have also compared the performance with prior tracking methods on KITTI and PETS datasets and observe an improvement of 7% in the number of successful tracks recovered, along with a speedup of >4X.

The rest of the paper is organized as follows: In Section II, we briefly survey the state-of-the-art works in traffic modeling, vehicle and pedestrian tracking. We describe our tracking algorithm, *DeepTAgent*, and HTMI in Section III. In Section IV, we present the tracking results on our dense traffic videos and also compare the accuracy with prior methods on other datasets.

II. Related Work

In this section, we give a brief overview of prior work on object tracking and motion modeling.

A. Pedestrian and Vehicle Tracking

There is extensive work on pedestrian tracking [1], [2]. Bruce et al. [3] and Gong et al. [4] predict pedestrians' motions by estimating their destinations. Liao et al. [5] compute a Voronoi graph from the environment and predicts pedestrian's motion along the edges. Mehran et al. [6] apply the social force model to detect people's abnormal behaviors from videos. Pellegrini et al. [7] use an energy function to build a goal-directed short-term collision-avoidance motion model. Bera et al. [8] use reciprocal velocity obstacles and hybrid motion models to improve the accuracy. All these methods are specifically designed for tracking pedestrian movement.

Vehicle tracking has been studied in computer vision, robotics, and intelligent transportation. Some of the earlier techniques are based on using cameras [9], [10], [11] and laser range finders [12], [13]. The recent developments in autonomous driving have resulted in the development of better sensors and new methods. [14] model dynamic and geometric properties of the tracked vehicles and estimate their positions. Using a stereo rig mounted on a mobile platform. [15] present an approach to detect and track vehicles in highly dynamic environments. [16], [17] use multiple cameras for tracking all surrounding vehicles. Moras et al. [18] use an occupancy grid framework to manage different sources of uncertainty for more efficient vehicle tracking, Wojke et al. [19] use LiDAR for moving vehicle detection and tracking in unstructured environments. [20] uses a feature-based approach to track the vehicles under varying lighting conditions. Most of these methods focus on vehicle tracking and do not take into account interactions with other traffic agents like pedestrians or bicycles in dense urban environments.

B. Motion Models and Tracking

There is substantial work on tracking multiple objects and use of motion models to improve the accuracy [21], [22], [23]. [21] presents an extension to MHT [24] so that it can be used with tracking-by-detection paradigm. [23] uses the constant velocity motion model to join fragmented pedestrian tracks caused by occlusion. RVO [25] is a non-linear motion model that has been used for pedestrian tracking in dense crowd videos. Recently, an extension to RVO has been proposed to model the trajectories of heterogeneous traffic agents with kinematic constraints [26]. Other motion models have been used for pedestrian tracking, including social forces [27], LTA [28], and ATTR [29].

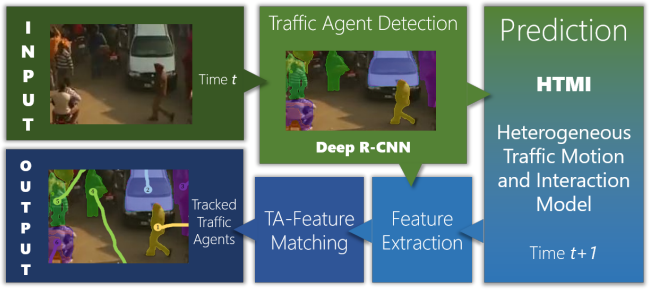

Fig. 2: *Overview of DeepTAgent: We use Mask R-CNN on an input frame at time t to generate segmented representations of agents. We use our novel motion and interaction model, HTMI, to predict the agent's state at frame $t+1$. We generate Deep TA-Features that are invariant to shape, size and scale of heterogeneous agents. These features are matched using association algorithms and a tracking ID is assigned to each predicted agent based on feature matching.*

C. Traffic Modeling and Navigation

Prior work in transportation engineering and robotics has focused on modeling the movement of vehicles and other road agents [30]. Traffic flow can be modeled using macroscopic [31], [32] or microscopic [33], [34], [35], [36] techniques. Luo et al. [37] propose a cellular automata model to simulate the car and bicycle heterogeneous traffic on an urban road. Chow et al. [38] model dynamic traffic based on the variational formulation of kinematic waves. Recent work in autonomous driving includes modeling human interactions or actions [39]. Our approach to model heterogeneous interactions is designed for dense traffic scenarios and complimentary to these approaches.

III. DeepTAgent: Tracking Heterogeneous Agents

In this section, we present our novel tracking algorithm that combines Mask R-CNN with a novel heterogeneous interaction model. One of the major challenges with tracking heterogeneous agents in dense traffic is that these agents corresponding to cars, buses, bicycles, pedestrians, etc. have different sizes, geometric shape, maneuverability, behavior, and dynamics (Figure 3). These often lead to complex inter-agent interactions that are usually not taken into account by prior multi-object trackers. For example, a bicyclist is riding next to a car, or a passenger is getting down from a bus. Furthermore, there are high-density scenarios, and these traffic-agents are in close-proximity or almost colliding

Fig. 3: *We show examples of common interactions that take place in dense traffic between heterogeneous agents. In this image, we observe two person-rickshaw interactions and one person-person interaction.*

configurations. So we need efficient techniques to predict their motion and interactions.

Overview: An overview of our approach is given in Fig. 2. All the symbols and notation used in the paper is highlighted in Table I. Our tracking algorithm starts by detecting all agents in $\mathcal{F}_t$ using Mask R-CNN as part of traffic agent detection. Mask R-CNN segments out the shape of each agent from its bounding box. These segmented representations consist of the exact shape of the agent overlaid on a white background. These representations are important for tracking accuracy. Furthermore, they are invariant to many factors such as shape, size, and scale [40], which are important in the context of heterogeneous traffic.

We use a novel heterogeneous traffic motion and interaction model (HTMI) to predict the position and velocity for each agent in $\mathbb{F}_{t+1}$. Our HTMI algorithm is used for two reasons: predicting the collision-free trajectory of each agent and capturing inter-agent interactions between different types of agents. This information is taken into account when computing the next state for an agent.

Finally, we use a simple feature extraction method, where we use a deep CNN to extract novel agent tracking features, called "Deep TA-Features," from both the predicted agents (positions and velocity) obtained from the HTMI model and detected segmented representations obtained from Mask R-CNN. Mask R-CNN creates a pyramid structure of the features called the Feature Pyramid Network (FPN), where it stacks features in layers. Each layer captures a different visual aspect of a traffic agent. For example, the bottom layer may capture the arms and legs of a pedestrian, and the topmost layer may correspond to facial features. With hundreds and thousands of layers in the FPN and each layer encoding some visual aspect like doors or tires, we generate the Deep TA-Features. These features are able to capture the heterogeneity in the traffic. For instance, a feature vector of a truck is easily distinguishable from a pedestrian. This facilitates easier matching of features of the truck in two successive frames, thereby, improving tracking accuracy. Finally, we match these features using the Cosine metric [41] and IOU overlap (percentage of bounding area overlap)

Symbol	Description
a_i	i^{th} agent
d_j	j^{th} detected agent (ie, it has a bounding box)
$\mathbb{F}_t$	current frame at timestep t
$\mathcal{A}$	set of all agents in the current frame, $\mathbb{F}_t$
$\mathcal{H}$	set of all detected agents in the current frame, $\mathbb{F}_t$
$\mathcal{H}_i$	subset of all detected agents in the current frame $\mathbb{F}_t$ that are within a circular region around agent a_i
$\mathbb{B}_{d_j}$	bounding box for detected agent d_j
$p_i \equiv (u_i, v_i)$	position of a_i, similarly defined for a_k
$\nu_i \equiv (\dot{u}_i, \dot{v}_i)$	velocity of a_i, similarly defined for a_k
f_{a_i}	feature vectors of the predicted agent, a_i,
f_{d_j}	feature vectors of the segmented representation, d_j
$l(p, q)$	cosine metric defined by $1 - p^T q$
$\psi_{t,i}$	state of an agent a_i at time t, includes position, velocity, and preferred velocity

TABLE I: *Notation and symbols used in the paper.*

measured based on the Hungarian algorithm [42]. The ID of the detected agent in $\mathbb{F}_t$ is matched with a predicted measurement in $\mathbb{F}_{t+1}$ and assigned to this agent.

A. Agent Detection Using Mask R-CNN

A key issue is to detect each different agent in the frame. The projections of the agents are shaped differently. Mask R-CNN uses a Feature Pyramid Network [40] that is ideal for heterogeneous agent tracking as it generates Deep TA-Features (we explain these features in detail in section III-C) that are invariant to multiple factors such as shape and size of heterogeneous agents. In addition, Mask R-CNN successfully detects agents in challenging environments such as night-time traffic with different lighting conditions, traffic with agents far away in the image. We also observe that R-CNN can also handle jittery camera motion (e.g., cameras mounted to an autonomous vehicle), and low-resolution images.

Our algorithm starts by detecting all the agents in the first frame. Mask R-CNN produces a set of bounding boxes, $\mathcal{B} = \{\mathbb{B}_{d_j} \mid \mathbb{B} = [\langle x, y\rangle_{\text{top left}}, w, h, s, r], d_j \in \mathcal{H}\}$, where $\langle x, y\rangle_{\text{top left}}, w, h, s,$ and r denote the top left corner, width, height, scale, and aspect ratio of $\mathbb{B}_{d_j}$, respectively.

Each bounding box has a corresponding mask for the agent it contains. A mask is a boolean array, each element of which is either true or false, depending on whether the pixel in that location belongs to the agent or not. We create a white background and super impose the pixel-wise segmented agent onto the background using the mask.

Let $\mathcal{W} = \{\mathbb{W}_{d_j}(\cdot) \mid d_j \in \mathcal{H}\}$ be the set of white canvases where each canvas, $\mathbb{W}_{d_j} = [\mathbb{1}]_{w \times h}$, w and h are the width and height of each $\mathbb{B}_{d_j}$ and $\mathbb{M}_{d_j}$ be the mask for d_j. Then,

$$\mathcal{U} = \{\mathbb{W}_{d_j}(\mathbb{M}_{d_j}) \mid \mathbb{W} \in \mathcal{W}, \mathbb{M} \in \mathcal{M}, d_j \in \mathcal{H}\},$$

is the set of segmented representations for each agent. In Proposition III.2, we show that appearance features generated from segmented representations outperform appearance features generated from bounding boxes that are produced by Faster R-CNN [43].

B. Heterogeneous Traffic Motion and Interaction (HTMI)

In order to track dense traffic, we need a model that takes into account the interaction between different types of agents with varying shape and dynamics. Moreover, high traffic density

Fig. 4: *An extremely dense traffic scene with 36 agents. We model complex inter-agent interactions between pedestrians, rickshaws, and cars for accurate tracking.*

increases the probability of collisions with other agents. In order to accurately track heterogeneous agents in dense videos, we need to take into account collision avoidance behavior as well as inter-agent interactions. We present a novel HTMI model that takes into account:

- Reciprocal collision avoidance [25] with car-like kinematic constraints for trajectory prediction and collision avoidance.
- Heterogeneous agent interaction that predicts when two or more agents will interact with each other in the near future.

1) Collision Avoidance: We represent each agent as, $\Psi_{t,i} = [u, v, \dot{u}, \dot{v}, v_{\text{pref}}]$, where $u, v, \dot{u}, \dot{v}$, and v_{pref} represent the top left corner of the bounding box, their velocities, and the preferred velocity respectively. The collision avoidance problem can be formally stated as $\Psi_{t+1,i} = f(\Psi_{t,i})$, where f is the RVO function that is used for velocity computation.

The computation of the new state, $\Psi_{t+1,i}$, is expressed as an optimization problem for each agent according to the RVO collision avoidance constraints [25]. If the RVO or ORCA constraints forbid an agent's preferred velocity, that agent chooses the closest velocity that lies in the feasible region:

$$\nu_{\text{RVO}} = \underset{v \notin ORCA}{\arg\max} ||v - \nu_{\text{pref}}|| \tag{1}$$

We can handle kinematic and dynamic constraints based on the control obstacles [44], which is a generalization to RVO formulation . This velocity, ν_{RVO}, is then used to calculate the new position of a traffic agent.

The difference in shapes, sizes, and aspect ratios of agents motivate the need to use appearance-based feature vectors as heterogeneity promotes differentiation between different agents on the basis of appearance. In order to combine object detection (R-CNN) with RVO, we modify the state vector, $\psi_{t,i}$, to include bounding box information by setting the position to the centers of the bounding boxes. Thus $u \leftarrow \frac{u+w}{2}, v \leftarrow \frac{v+h}{2}$.

2) Heterogeneous Traffic Interactions: When two traffic agents interact, they move towards each other and come in close proximity. We assume that two agents intend to interact if they have been moving towards each other for some time τ. We now present a condition for two agents a_i, a_k to

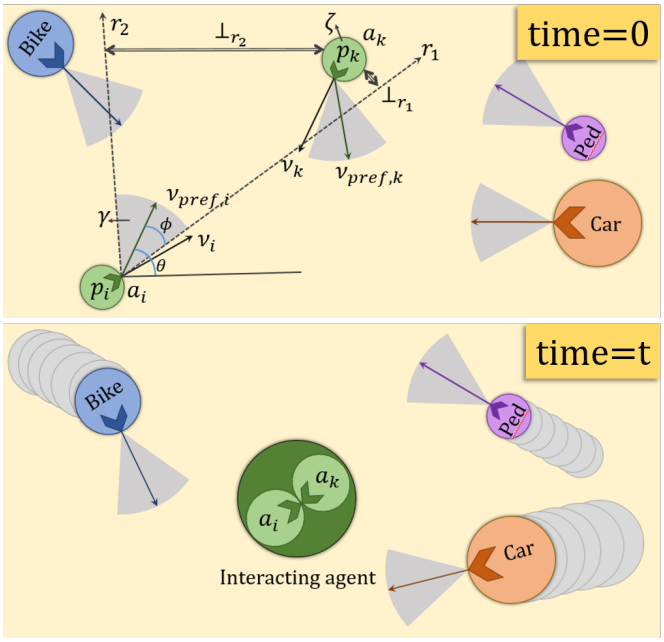

Fig. 5: *(top) At time $t = 0$, we predict that a strong condition to determine possible interaction between a_i and a_k is if γ (grey cone) of a_i intersects ζ (green circle around a_k) or envelopes it. The first is mathematically equivalent to r_1 or r_2 intersecting ζ (top). The second condition is equivalent to ζ lying inside the projected cone of a_i. (bottom.) In the event of an interaction between a_i and a_k, we assume that the interacting pair as a new agent and model its kinematics accordingly.*

be able to interact and use this information as part of our prediction module (see Figure 5). Given a traffic agent a_i, the slope of $\nu_{\text{pref},i}$ is $\tan\theta_i$. In dense traffic, each agent has a limited space in which they can steer, or turn. We denote this steering angle as ϕ_i. We define a circular region, ζ of radius ρ, centered around a_k that represents the personal space of a_k. Based on the direction of current preferred velocity and the steering angle, the space in which a_i can freely move is defined by a 2D cone, which we denote as γ. We denote the extreme rays of γ as r_1 and r_2. $\perp_{g_1}^{g_2}$ denotes the smallest perpendicular distance between two geometric structures, g_1 and g_2.

Our heterogeneous interaction module assumes that if the *projected cone* of a_i, defined by extending r_1 and r_2, intersects with the ζ, then a_i can interact with a_k. Based on this assumption, it is sufficient to check for either one of two conditions: intersection of ζ with either r_1 or r_2 (if either ray intersects, then the entire cone intersects ζ) and $\zeta \subset \gamma$ (if ζ lies in the interior of γ, see Figure 5). r_1, r_2 are parameterized based on their slopes $\tan\delta$, where $\delta = \theta_i + \phi_i$ if $\perp_\zeta^{r_1} \geq \perp_\zeta^{r_2}$, else $\delta = \theta_i - \phi_i$. The resulting equation of r_1 (or r_2) is $(Y - v_i) = \tan\delta(X - u_i)$ and the equation of ζ is $(X - u_k)^2 + (Y - v_k)^2 = \rho^2$. Solving both equations, we obtain

$$X^2\big(\sec^2\delta\big) + 2X\big(\tan\delta(v_i - v_k) - u_k - u_i\tan^2\delta\big) + \big(u_k^2 + (v_i - v_k)^2 + u_i^2\tan^2\delta - 2\tan\delta(v_i - v_k)u_i - \rho^2\big), \tag{2}$$

Intersection occurs if the discriminant, Ω_1, of (2) ≥ 0. This provides us with one condition for the occurrence of an interaction between a_i and a_k.

Moreover, we observe that if ζ lies in the interior of γ, then p_k lies on the opposite sides of r_1 and r_2 which is modeled by the following equation:

$$r_1(p_k).r_2(p_k) \leq 0 \tag{3}$$

Solving (3) further provides us the second condition for the occurrence of an interaction between a_i and a_k,

$$\Omega_2 \equiv (v_k - v_i)^2 + (u_k - u_i)^2 \tan\alpha \tan\beta$$
$$- (v_k - v_i)(u_k - u_i)(\tan 2\theta_i)(1 - \tan\alpha \tan\beta) \leq 0$$

where $\Omega_1, \Omega_2 : \mathbb{R}^2 \times \mathbb{R}^2 \times \mathbb{R} \times \mathbb{R} \longmapsto \mathbb{R}$.

If either Ω_1 or Ω_2 is true, then agents a_i, a_k will move towards each other to interact in the near future. When this happens, we assume that a_i and a_k align their current velocities towards each other. In that case, the time taken for the two agents to be very close (or overlap) with each other is given by:

$$t = \frac{||p_i - p_k||_2}{||\nu_i - \nu_k||_2}$$

If two agents are overlapping (based on the values of Ω_1 and Ω_2), we model them as a new agent with radius 2ϵ. In this case, two agents are classified to be interacting if, $||p_i - p_k||_2 = \epsilon$. Our approach can be extended to model interactions. We form a set $\mathcal{I} \subseteq \mathcal{A}$, where $\mathcal{I}$ is the set of all agents a_ω, that are intending to interact with a_k. In order to determine the first agent that may interact with a_k, we can compute the time taken by that agent that wishes to interact with a_k by: $t_{\text{min}} = \arg\min_t ||\nu_\omega t - p_k|| = \rho, a_\omega \in \mathcal{I}$. Agents that are not interacting avoid each other and continue moving towards their destination.

We use our model to formulate when two agents can interact in a densely crowded traffic scene. We model the time taken by that agent to reach the other agent, and we also take into account multiple interactions in the same scene. Using the new positions and velocities after capturing these interactions, HTMI uses RVO to predict new positions and velocities of the interacting agents.

C. Feature Extraction Using Segmented Representations

We extract simple, yet powerful, features, called "Deep TA-Features", from segmented shapes, that are obtained from Mask R-CNN, using a convolutional neural network. The architectural and training details of this network are described in [45]. Deep TA-Features are grid-like maps that are stacked in a pyramid-shaped structure. Each layer of this pyramid encodes a visual aspect (for example, facial features or vehicle parts) of a traffic agent. We present a formulation in which Deep TA-Features significantly improve the performance of our algorithm by reducing false negatives, (ground truth agents that are not tracked) and identity switches (incorrect assignment of tracking ID). These often arise in dense scenarios due to increased instances of collisions and interactions.

We define $\mathcal{T}_{t,i} = \{\Psi_{1:t,i}\}$ to be the set of states for a correctly tracked traffic agent, a_i until time t. We denote the time since the last update to an agent's ID as t_{lu}. We denote the ID of a_i as α_i and we represent the correct assignment of an ID to a_i as $\Gamma(\alpha_i)$. The threshold for the cosine distance is $\lambda \underset{\text{i.i.d.}}{\sim} \mathbb{U}[0,1]$. The threshold for the track age is t_a. We denote the probability of an event that uses Mask R-CNN by $\mathbb{P}^M(\cdot)$ and the probability of an event that uses Faster R-CNN by $\mathbb{P}^F(\cdot)$. $\mathbb{P}$ without a superscript denotes general probability. Finally, $\mathcal{T}_{t,i} \leftarrow \{\emptyset\}$ represents the loss of $\mathcal{T}_{t,i}$ by occlusion, collision, or interaction between heterogeneous agents in dense traffic.

The following lemma highlights some properties of these feature vectors. It shows that the Cosine cost of Deep TA-Features is less than the Cosine cost of features generated from prior algorithms like Faster R-CNN. Thus,

Lemma III.1. *For every pair of feature vectors, $(f^M_{d_j}, f^F_{d_j})$ generated from a segmented box and a bounding box, respectively, $l(f_{a_i}, f^M_{d_j}) < l(f_{d_j}, f^F_{d_j})$.*

The proof is described in the report [46].

Proposition III.2. *Feature vectors extracted from segmented representations decrease the probability of the loss of agent tracks and increase the probability of correct ID assignment to an agent, without additional training or complex optimizations, thereby reducing the number of false negatives and ID switches.*

Proof: The correct assignment of an ID depends on successful feature matching between the predicted measurement feature and the optimal segmented shape feature. Equivalently,

$$d(f_{a_i}, f_{h^*_{j,a_i}}) > \lambda \Leftrightarrow \alpha_i = \emptyset, \forall a_i. \tag{4}$$

Using III.1 and the fact that $\lambda \underset{i.i.d.}{\sim} \mathbb{U}[0,1]$,

$$\mathbb{P}(d(f_{a_i}, f^M_{h^*_{j,a_i}}) > \lambda) < \mathbb{P}(d(f_{a_i}, f^F_{h^*_{j,a_i}}) > \lambda)$$

Using (4), it directly follows that $\mathbb{P}^M(\alpha_i = \emptyset) < \mathbb{P}^F(\alpha_i = \emptyset)$. In our approach, we set $t_{\text{lu}} > t_a \wedge \alpha_i = \emptyset \Leftrightarrow \mathcal{T}_{t,i} \leftarrow \{\emptyset\}$. It follows that,

$$\mathbb{P}^M(\mathcal{T}_{t,i} \leftarrow \{\emptyset\}) < \mathbb{P}^F(\mathcal{T}_{t,i} \leftarrow \{\emptyset\}). \tag{5}$$

We define the total number of false negatives (FN) as $FN = \sum_{t=1}^{T} \sum_{a_g \in \mathcal{G}} \delta_{\mathcal{T}_{t,a_g}}$, where $a_g \in \mathcal{G}$ denotes a ground truth agent in the set of all ground truth agents in $\mathbb{F}_t$ and $\delta_z = 1$ for $z = 0$ and 0 elsewhere, is a variation of the Kronecker delta function. Using (5), we can say that fewer lost tracks ($\mathcal{T}_{t,i} \leftarrow \{\emptyset\}$) indicate a smaller number of false negatives. Finally, by using lemma III.1 with the association formulation (6) described in the next subsection, it is easy to see that $\mathbb{P}^M(\Gamma(\alpha_i)) > \mathbb{P}^F(\Gamma(\alpha_i))$, which completes the proof. ■

D. Feature Matching Using Association Algorithms

Deep TA-Features corresponding to the same agent ID are likely to be more similar, and we use appropriate algorithms.

We measure this similarity in two ways: the Cosine metric [41] and the IOU overlap [47]. The task of assigning a new ID to a predicted measurement for a_i becomes that of computing the optimum detection, $d^*_{j,i}$; that is, the detection whose feature vector is most similar to f_{a_i}. This is posed as an optimization problem:

$$d^*_{j,a_i} = \arg\min_{d_j}(l(f_{a_i}, f_{d_j})|a_i \in \mathcal{A}, d_j \in \mathcal{H}_i). \quad (6)$$

The IOU overlap metric is used together with the cosine metric. This metric builds a cost matrix, C to measure the amount of overlap of each predicted bounding box with all nearby detection bounding box candidates. $C(i,j)$ stores the IOU overlap of the bounding box of Ψ_{t+1,a_i} with that of d_j and is calculated by

$$C(i,j) = \frac{\mathbb{B}_{\Psi_{t+1,i}} \cap \mathbb{B}_{d_j}}{\mathbb{B}_{\Psi_{t+1,i}} \cup \mathbb{B}_{d_j}}, d_j \in \mathcal{H}_i.$$

Matching a detection to a predicted measurement with maximum overlap thus becomes a max weight matching problem that can be solved using the Hungarian algorithm [42].

IV. Implementation and Results

In this section, describe our implementation and highlight the performance on different datasets.

A. New Dataset and Baseline Results

We use a new dataset that consists of a set of 19 video sequences that contain dense traffic with highly heterogeneous agents with varying viewpoints, camera motions, and at different times of the day. These videos correspond to the highway and urban traffic in the USA, China, and India. Most importantly, ground truth annotations are provided with the dataset.

The key aspects of this dataset are the density and the degree of heterogeneity. Compared to standard traffic datasets such as KITTI [48] and PETS[49], our dataset is denser with a higher degree of heterogeneity (table IV. This includes challenging sequences such as nighttime traffic with heavy glare from oncoming traffic, scenes with jittery camera motion, and scenes with agents far away from the camera.

We provide baseline results on our dataset using our tracking algorithm and demonstrate a high average accuracy of 72.02% (table III) operating at an average speed of 0.4s on a single GPU. However, our algorithm can be optimized further by sub-sampling and using parallelization to increase the frame rate. We classify accuracy according to the CLEAR metrics [50] as $1-(FN+FP+IDSW)/GT$, where FN, FP, IDSW, and GT correspond to the false negatives, false positives, ID switches, and ground truth, respectively. We do not count stationary agents such as parked vehicles in our formulation. Objects such as traffic signals are considered false positives. We take into account all possible road agents. Thus, negatives that include agents such as carts and animals are also considered as false negatives.

B. Comparison: Online Methods on Standard Benchmarks

We also evaluate our tracking algorithm on the KITTI-16 sequence and compare with online methods on the MOT benchmark [51]. We only compare our algorithm with methods that have an average rank higher than ours on the MOT benchmark. We achieve the lowest number of false negatives and identity switches, which is a direct consequence of III.2.

V. Conclusion and Future Work

We present a realtime, novel, end-to-end tracking algorithm called DeepTAgents for traffic agents, including vehcles, bicycles, pedestrians, etc. in highly dense scenarios. We use tracking-by-detection paradigm and present a new motion and interaction model (HTMI). Our approach is general and evaluated on a new dense video datasets. Our approach is $4X$ faster than prior tracking algorithms and we observe considerable improvement in accuracy.

There are many avenues for future work. Besides accelerating the performance, we can further improve the accuracy. This includes improved HTMI model, that uses a more accurate model for dynamics as well as improving the accuracy of TA-feature detection. We would like to further evaluate the performance on complex traffic videos.

References

[1] J. Cui, H. Zha, H. Zhao, and R. Shibasaki, "Tracking multiple people using laser and vision," in *Proc. of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2005, pp. 2116–2121.

[2] L. Kratz and K. Nishino, "Tracking pedestrians using local spatio-temporal motion patterns in extremely crowded scenes," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, no. 99, p. 11, 2011.

[3] A. Bruce and G. Gordon, "Better motion prediction for people-tracking," in *Proc. of the International Conference on Robotics and Automation (ICRA), New Orleans, USA*, 2004.

[4] H. Gong, J. Sim, M. Likhachev, and J. Shi, "Multi-hypothesis motion planning for visual object tracking," pp. 619–626, 2011.

[5] L. Liao, D. Fox, J. Hightower, H. Kautz, and D. Schulz, "Voronoi tracking: Location estimation using sparse and noisy sensor data," in *IROS*, 2003.

[6] R. Mehran, A. Oyama, and M. Shah, "Abnormal crowd behavior detection using social force model," in *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition,CVPR*, 2009, pp. 935–942.

[7] S. Pellegrini, A. Ess, K. Schindler, and L. Van Gool, "You'll never walk alone: Modeling social behavior for multi-target tracking," in *ICCV*, 2009.

[8] A. Bera and D. Manocha, "REACH: Realtime crowd tracking using a hybrid motion model," *ICRA*, 2015.

[9] T. Zielke, M. Brauckmann, and W. Von Seelen, "Intensity and edge-based symmetry detection with an application to car-following," *CVGIP Image Understanding*, vol. 58, pp. 177–177, 1993.

[10] E. D. Dickmanns, "Vehicles capable of dynamic vision: a new breed of technical beings?" *Artificial Intelligence*, vol. 103, no. 1-2, pp. 49–76, 1998.

[11] F. Dellaert and C. Thorpe, "Robust car tracking using kalman filtering and bayesian templates," in *Conference on intelligent transportation systems*, vol. 1, 1997.

Dataset	Tracker	FPS↑	FN↓	MT(%)↑	ML(%)↓	MOTA(%)↑	IDSW↓
KITTI-16	AP_HWDPL_p	6.7	831	17.6	11.8	40.7	18
	RAR_15_pub	5.4	809	0.0	17.6	41.2	18
	AMIR15	1.9	714	11.8	11.8	50.4	18
	HybridDAT	4.6	706	5.9	17.6	46.3	10
	DeepTAgent	**28.9**	**668**	**29.4**	**11.7**	12.2	15

TABLE II: *Evaluation on the KITTI-16 benchmark with online methods that use public detections that have an average rank higher than ours. Bold is best, blue is second best. References for these papers can be found at* `https://motchallenge.net/`. *Arrows (↑,↓) indicate the direction of better performance. Our algorithm is >4X faster and 7% more accurate in the number of correctly tracked agents*

Sequence	Length	Day/Night	Camera Motion	Accuracy(MOTA)
TRAF12	0:35	Day	Moving	73.27%
TRAF15	0:29	Day	Moving	68.34%
TRAF17	0:19	Day	Moving	65.94%
TRAF18	0:33	Day	Moving	77.50%
TRAF19	0:17	Day	Moving	67.15%
TRAF20	1:00	Day	Moving	81.32%
TRAF21	0:31	Day	Moving	79.92%
TRAF22	0:45	Day	Moving	66.87%
TRAF23	0:19	Night	Moving	67.92%

TABLE III: *We highlight the tracking accuracy of DeepTAgent on **9** of our 19 high resolution (720p) video sequences of dense heterogeneous traffic with different types of agents. All sequences have been carefully annotated with ground truth labels following a strict protocol. The sequences are categorized according to viewpoint, time of the day, camera motion, and length. Tracking accuracy on all the 19 sequences are available on our website.*

Dataset (# classes)	Person	Car	Bike	Scooter	Bus	Truck	Cycle	Rickshaw	Animal
KITTI(3)	9.5	2	0	0	0	0	0	1.5	0
MOT (3)	13	3	0	0	0.5	0	0	0	0
PETS09 (2)	13	0	0	0	0	0.5	0	0	0
Our Dataset (8)	**28.5**	**37**	**22**	**8**	**2**	**0.5**	**1.5**	**14.5**	**1**

TABLE IV: *We pick the top two sequences of various traffic datasets and compare heterogeneity with ours (TRAF11 and TRAF12). We observe a high variety of traffic agents.*

[12] L. Zhao and C. Thorpe, "Qualitative and quantitative car tracking from a range image sequence," in *Computer Vision and Pattern Recognition, 1998. Proceedings. 1998 IEEE Computer Society Conference on*. IEEE, 1998, pp. 496–501.

[13] D. Streller, K. Furstenberg, and K. Dietmayer, "Vehicle and object models for robust tracking in traffic scenes using laser range images," in *Intelligent Transportation Systems, 2002. Proceedings. The IEEE 5th International Conference on*. IEEE, 2002, pp. 118–123.

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

[15] A. Ess, K. Schindler, B. Leibe, and L. Van Gool, "Object detection and tracking for autonomous navigation in dynamic environments," *The International Journal of Robotics Research*, vol. 29, no. 14, pp. 1707–1725, 2010.

[16] A. Rangesh and M. M. Trivedi, "No blind spots: Full-surround multi-object tracking for autonomous vehicles using cameras & lidars," *arXiv preprint arXiv:1802.08755*, 2018.

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

[18] J. Moras, V. Cherfaoui, and P. Bonnifait, "Credibilist occupancy grids for vehicle perception in dynamic environments," in *Robotics and Automation (ICRA), 2011 IEEE International Conference on*. IEEE, 2011, pp. 84–89.

[19] N. Wojke and M. Häselich, "Moving vehicle detection and tracking in unstructured environments," in *Robotics and Automation (ICRA), 2012 IEEE International Conference on*. IEEE, 2012, pp. 3082–3087.

[20] B. Coifman, D. Beymer, P. McLauchlan, and J. Malik, "A real-time computer vision system for vehicle tracking and traffic surveillance," *Transportation Research Part C: Emerging Technologies*, vol. 6, no. 4, pp. 271–288, 1998.

[21] C. Kim, F. Li, A. Ciptadi, and J. M. Rehg, "Multiple hypothesis tracking revisited," in *Proceedings of the IEEE International Conference on Computer Vision*, 2015, pp. 4696–4704.

[22] J. Chen, H. Sheng, Y. Zhang, and Z. Xiong, "Enhancing detection model for multiple hypothesis tracking," in *Conf. on Computer Vision and Pattern Recognition Workshops*, 2017, pp. 2143–2152.

[23] H. Sheng, L. Hao, J. Chen, Y. Zhang, and W. Ke, "Robust local effective matching model for multi-target tracking," in *Pacific Rim Conference on Multimedia*. Springer, 2017, pp. 233–243.

[24] D. Reid *et al.*, "An algorithm for tracking multiple targets," *IEEE transactions on Automatic Control*, vol. 24, no. 6, pp. 843–854, 1979.

[25] J. Van Den Berg, S. J. Guy, M. Lin, and D. Manocha, "Reciprocal n-body collision avoidance," in *Robotics research*. Springer, 2011, pp. 3–19.

[26] Y. Ma, D. Manocha, and W. Wang, "Autorvo: Local navigation with dynamic constraints in dense heterogeneous traffic," *arXiv preprint arXiv:1804.02915*, 2018.

[27] A. Bera, N. Galoppo, D. Sharlet, A. Lake, and D. Manocha, "Adapt: real-time adaptive pedestrian tracking for crowded scenes," in *Robotics and Automation (ICRA), 2014 IEEE International Conference on*. IEEE, 2014, pp. 1801–1808.

[28] S. Pellegrini, A. Ess, K. Schindler, and L. van Gool, "You'll never walk alone: Modeling social behavior for multi-target tracking," in *2009 IEEE 12th International Conference on Computer Vision*, Sept 2009, pp. 261–268.

[29] K. Yamaguchi, A. C. Berg, L. E. Ortiz, and T. L. Berg, "Who are you with and where are you going?" in *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*. IEEE, 2011, pp. 1345–1352.

[30] B. Paden, M. Čáp, S. Z. Yong, D. Yershov, and E. Frazzoli, "A survey of motion planning and control techniques for self-driving urban vehicles," *IEEE Transactions on intelligent vehicles*, vol. 1, no. 1, pp. 33–55, 2016.

[31] J.-P. Lebacque, "First-order macroscopic traffic flow models: Intersection modeling, network modeling," in *Transportation and Traffic Theory. Flow, Dynamics and Human Interaction. 16th International Symposium on Transportation and Traffic TheoryUniversity of Maryland, College Park*, 2005.

[32] N. Geroliminis, C. F. Daganzo *et al.*, "Macroscopic modeling of traffic in cities," in *86th Annual Meeting of the Transportation Research Board, Washington, DC*, 2007.

[33] B. S. Kerner, S. L. Klenov, and D. E. Wolf, "Cellular automata approach to three-phase traffic theory," *Journal of Physics A: Mathematical and General*, vol. 35, no. 47, p. 9971, 2002.

[34] A. Schadschneider, "Traffic flow: a statistical physics point of view," *Physica A: Statistical Mechanics and its Applications*, vol. 313, no. 1-2, pp. 153–187, 2002.

[35] S. Maerivoet and B. De Moor, "Cellular automata models of road traffic," *Physics Reports*, vol. 419, no. 1, pp. 1–64, 2005.

[36] G. Pandey, K. R. Rao, and D. Mohan, "A review of cellular automata model for heterogeneous traffic conditions," in *Traffic and Granular Flow'13*. Springer, 2015, pp. 471–478.

[37] Y. Luo, B. Jia, J. Liu, W. H. Lam, X. Li, and Z. Gao, "Modeling the interactions between car and bicycle in heterogeneous traffic," *Journal of advanced transportation*, vol. 49, no. 1, pp. 29–47, 2015.

[38] A. H. Chow, S. Li, W. Szeto, and D. Z. Wang, "Modelling urban traffic dynamics based upon the variational formulation of kinematic waves," *Transportmetrica B: Transport Dynamics*, vol. 3, no. 3, pp. 169–191, 2015.

[39] D. Sadigh, S. Sastry, S. A. Seshia, and A. D. Dragan, "Planning for autonomous cars that leverage effects on human actions," in *Robotics: Science and Systems*, 2016.

[40] K. He, G. Gkioxari, P. Dollár, and R. Girshick, "Mask R-CNN," *ArXiv e-prints*, Mar. 2017.

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

[42] H. W. Kuhn, "The hungarian method for the assignment problem," in *50 Years of Integer Programming 1958-2008*. Springer, 2010, pp. 29–47.

[43] Y. Zhang, J. Wang, and X. Yang, "Real-time vehicle detection and tracking in video based on faster r-cnn," in *Journal of Physics: Conference Series*, vol. 887, no. 1. IOP Publishing, 2017, p. 012068.

[44] D. Bareiss and J. van den Berg, "Generalized reciprocal collision avoidance," *The International Journal of Robotics Research*, vol. 34, no. 12, pp. 1501–1514, 2015.

[45] N. Wojke, A. Bewley, and D. Paulus, "Simple Online and Realtime Tracking with a Deep Association Metric," *ArXiv e-prints*, Mar. 2017.

[46] R. Chandra, T. Randhavane, U. Bhattacharya, A. Bera, and D. Manocha, "Deeptagent: Realtime tracking of dense traffic agents using heterogeneous interaction," *Technical Report*, 2018. [Online]. Available: http://gamma.cs.unc.edu/HTI/DeepTAgent.pdf

[47] M. Levandowsky and D. Winter, "Distance between sets," *Nature*, vol. 234, no. 5323, p. 34, 1971.

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

[49] J. Ferryman and A. Shahrokni, "Pets2009: Dataset and challenge," in *2009 Twelfth IEEE International Workshop on Performance Evaluation of Tracking and Surveillance*. IEEE, 2009, pp. 1–6.

[50] A. W. Smeulders, D. M. Chu, R. Cucchiara, S. Calderara, A. Dehghan, and M. Shah, "Visual tracking: An experimental survey," *IEEE Transactions on Pattern Analysis & Machine Intelligence*, no. 1, p. 1, 2013.

[51] A. Milan, L. Leal-Taixé, I. Reid, S. Roth, and K. Schindler, "Mot16: A benchmark for multi-object tracking," *arXiv preprint arXiv:1603.00831*, 2016.