DensePeds: Pedestrian Tracking in Dense Crowds Using Front-RVO and Sparse Features

Rohan Chandra[1], Uttaran Bhattacharya[1], Aniket Bera[2], and Dinesh Manocha[1]
[1]University of Maryland, [2]University of North Carolina
https://gamma.umd.edu/ad/densepeds

***Abstract*— We present a pedestrian tracking algorithm, *DensePeds*, that tracks individuals in highly dense crowds (>2 pedestrians per square meter). Our approach is designed for videos captured from front-facing or elevated cameras. We present a new motion model called Front-RVO (FRVO) for predicting pedestrian movements in dense situations using collision avoidance constraints and combine it with state-of-the-art Mask R-CNN to compute sparse feature vectors that reduce the loss of pedestrian tracks (false negatives). We evaluate DensePeds on the standard MOT benchmarks as well as a new dense crowd dataset. In practice, our approach is 4.5× faster than prior tracking algorithms on the MOT benchmark and we are state-of-the-art in dense crowd videos by over 2.6% on the absolute scale on average.**

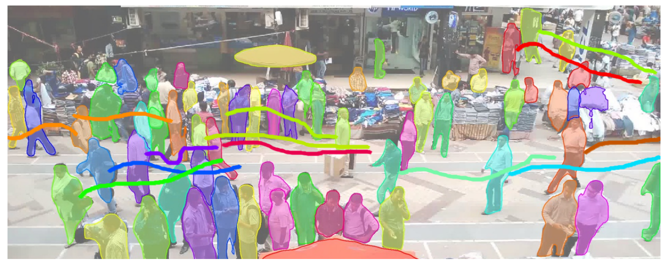

Fig. 1. Performance of our pedestrian tracking algorithm on the NPLACE-1 sequence with over 80 pedestrians per frame. The colored tracks associated with each moving pedestrian are linked to a unique ID. DensePeds achieves an accuracy of up to 85.5% on our new dense crowd dataset and improves tracking accuracy by 2.6% over state of the art methods on the average. This is equivalent to an average rank difference of 17 on the MOT benchmark.

I. Introduction

Pedestrian tracking is the problem of maintaining the consistency in the temporal and spatial identity of a person in an image sequence or a crowd video. This is an important problem that helps us not only extract trajectory information from a crowd scene video but also helps us understand high-level pedestrian behaviors [1]. Many applications in robotics and computer vision such as action recognition and collision-free navigation and trajectory prediction [2] require tracking algorithms to work accurately in real time [3]. Furthermore, it is crucial to develop general-purpose algorithms that can handle front-facing cameras (used for robot navigation or autonomous driving) as well as elevated cameras (used for urban surveillance), especially in densely crowded areas such as airports, railway terminals, or shopping complexes.

Closely related to pedestrian tracking is pedestrian detection, which is the problem of detecting multiple individuals in each frame of a video. Pedestrian detection has received a lot of traction and gained significant progress in recent years. However, earlier work in pedestrian tracking [4], [5] did not include pedestrian detection. These tracking methods require manual, near-optimal initialization of each pedestrian's state information in the first video frame. Further, sans-detection methods need to know the number of pedestrians in each frame apriori, so they do not handle cases in which new pedestrians enter the scene during the video. Tracking by detection overcomes these limitations by employing a detection framework to recognize pedestrians entering at any point of time during the video and automatically initialize their state-space information.

However, tracking pedestrians in dense crowd videos where there are 2 or more pedestrians per square meter remains a challenge for the tracking-by-detection literature. These videos suffer from severe occlusion, mainly due to the pedestrians walking extremely close to each other and frequently crossing each other's paths. This makes it difficult to track each pedestrian across the video frames. Tracking-by-detection algorithms compute a bounding box around each pedestrian. Because of their proximity in dense crowds, the bounding boxes of nearby pedestrians overlap which affects the accuracy of tracking algorithms.

Main Contributions. Our main contributions in this work are threefold.

1) We present a pedestrian tracking algorithm called DensePeds that can efficiently track pedestrians in crowds with 2 or more pedestrians per square meter. We call such crowds "dense". We introduce a new motion model called Frontal Reciprocal Velocity Obstacles (FRVO), which extends traditional RVO [6] to work with front- or elevated-view cameras. FRVO uses an elliptical approximation for each pedestrian and estimates pedestrian dynamics (position and velocity) in dense crowds by considering intermediate goals and collision avoidance constraints.
2) We combine FRVO with the state-of-the-art Mask R-CNN object detector [7] to compute sparse feature vectors. We show analytically that our sparse feature vector computation reduces the probability of the loss of pedestrian tracks (false negatives) in dense scenarios. We also show by experimentation that using sparse feature vectors makes our method outperform state-of-the-art tracking methods on dense crowd videos, without any additional requirement for training or optimization.
3) We make available a new dataset consisting of dense crowd videos. Available benchmark crowd videos are extremely sparse (less than 0.2 persons per square meter). Our dataset, on the other hand, presents a more challenging and realistic view of crowds in public areas in dense metropolitans.

On our dense crowd videos dataset, our method outperforms the state-of-the-art, by 2.6% and reduce false negatives by 11% on our dataset. We also validate the benefits of FRVO and sparse feature vectors through ablation experiments on our dataset. Lastly, we evaluate and compare the performance of DensePeds with state-of-the-art online methods on the MOT benchmark [8] for a comprehensive evaluation. The benefits of our method do not hold for sparsely crowded videos such as the ones in the MOT benchmark since FRVO is based on reciprocal velocities of two colliding pedestrians. Unsurprisingly, we do not outperform the state-of-the-art methods on the MOT benchmark since the MOT sequences do not contain many colliding pedestrians, thereby rendering FRVO ineffective. Nevertheless, our method still has the lowest false negatives among all the available methods. Its overall performance is state of the art on dense videos and is in the top 13% of all published methods on the MOT15 and top 20% on MOT16.

II. Related Work

The most relevant prior work for our work is on object detection, pedestrian tracking, and motion models used in pedestrian tracking. Each of these fields is well-studied, and a comprehensive overview for each of them is well beyond the scope of this paper. We only present a brief overview of prior work on these fields that are most closely related to our work in this paper.

A. Object Detection

Early methods for object detection include HOG [9] and SIFT [10], which manually extract features from images. Inspired by AlexNet, [11] proposed R-CNN for optimizing the object detection problem by incorporating a selective search. Subsequently, Fast R-CNN [12] and Faster R-CNN [13] were proposed to improve the speed of the original R-CNN.

More recently, the prevalence of CNNs has led to the development of Mask R-CNN [7], which extends Faster R-CNN to include pixel-level segmentation. The use of Mask R-CNN in pedestrian tracking has been limited, although it has been used for other pedestrian-related problems such as pose estimation [14].

B. Pedestrian Tracking

There have been considerable advances in object detection since the advent of to deep learning, which has led to substantial research in the intersection of deep learning and the tracking-by-detection paradigm [15], [16], [17], [18]. However, [18], [16] operate at less than one fps on a GPU and have low accuracy on standard benchmarks while [15] sacrifices accuracy to increase tracking speed. Finally, deep learning methods require expensive computation, often prohibiting real-time performance with inexpensive computing resources. Some recent methods [19], [20] achieve high accuracy but may require high-quality, heavily optimized detection features for good performance. For an up-to-date review of tracking-by-detection algorithms, we refer the reader to methods submitted to the MOT [8] benchmark.

C. Motion Models in Pedestrian Tracking

Motion models are commonly used in pedestrian tracking algorithms to improve tracking accuracy [21], [22], [23], [24]. [22] presents a variation of MHT [25] and shows that it is at par with the state-of-the-art from the tracking-by-detection paradigm. [24] uses a motion model to combine fragmented pedestrian tracks caused by occlusion. These methods are based on linear constant velocity or acceleration models. Such linear models, however, cannot characterize pedestrian dynamics in dense crowds [26]. RVO [6] is a non-linear motion model that has been used for pedestrian tracking in dense videos to compute intermediate goal locations, but it only works with top-facing videos and circular pedestrian representations. Other motion models used in pedestrian tracking are the Social Force model [27], LTA [28], and ATTR [29].

There are also many discrete motion models that represent each individual or pedestrian in a crowd as a particle (or as a 2D circle on a plane) to model the interactions. These include models based on simple interaction rules [30], repulsive forces [31] and velocity-based optimization algorithms [32], [6]. More recent discrete approaches are based on short-term planning using a discrete approach [33] and cognitive models [34]. However, these methods are based on circular agent representation and do not work well for front-facing pedestrians in dense crowd videos as they are overly conservative in terms of motion prediction.

III. FRVO: Local Trajectory Prediction

When walking in dense crowds, pedestrians keep changing their velocities frequently to avoid collisions with other pedestrians. Pedestrians also exhibit local interactions and other collision-avoidance behaviors such as side-stepping, shoulder-turning, and backpedaling [35]. As a result, prior motion models with constant velocity or constant acceleration assumptions do not accurately model crowded scenarios. Further, defining an accurate motion model gets even more challenging in front-view videos due to occlusion and proximity issues. For example, in top-view videos, one uses circular representations to model the shape of the heads of pedestrians. It is, generally, physically impossible for two individual's heads to occlude or occupy the same space. In front facing crowd videos, however, occlusion is a common problem. In this section, we introduce our non-linear motion model, Frontal RVO (FRVO), which is designed to work with crowd videos captured using front- or elevated-view cameras.

A. Notations

Table I lists the notations used in this paper. A few important points to keep in mind are:

- While h_j denotes a detected pedestrian, f_{h_j} denotes the feature vector extracted from the *segmented box* of h_j.
- $\mathcal{H}_i = \{h_j \ : \ |p_i - h_j| \leq \rho, p_i \in \mathcal{P}\}$

B. Analytical Comparison with RVO

Our motion model is an extension of the RVO (Reciprocal Velocity Obstacle) [6] approach, which can accurately model the trajectories of pedestrians using collision avoidance constraints. However, RVO models pedestrians using circular shapes which result in false overlaps in front- and elevated-view camera videos (Figure 3). A false overlap is essentially a false positive wherein RVO would signal a collision while the actual scene would not contain a collision. We show here that false overlaps cause the accuracy of RVO to drop by an error margin of Δ.

p_i	i^{th} pedestrian
h_j	j^{th} detected pedestrian
f_{p_i} & f_{h_j}	feature vectors corresponding to p_i and h_j respectively
T	total number of frames in the video
$\mathcal{H}$ & $\mathcal{P}$	sets of all pedestrian detections and total pedestrians in a frame respectively
ρ	predefined radius around every p_i
$\mathcal{H}_i$	set of all detected persons within a circle of radius r centered around p_i
N_i	cardinality of the set $\mathcal{H}_i$
$d(a,b)$	cosine distance metric between the vectors a and b, defined as: $d(a,b) = 1 - \frac{a^\top b}{\|a\|\|b\|}$

TABLE I
NOTATIONS WE USE IN THIS PAPER.

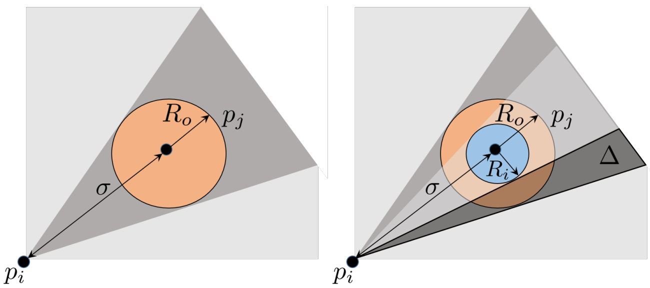

Fig. 2. (Left) Standard VO configuration fundamental to RVO, using circular representations for top-view. (Right) RVO for front and elevated videos. The cyan circle and it's corresponding transparent gray cone represent the VO configuration for δ-overlapping circles. RVO considers the highlighted area of the dark gray cone to be part of the collision area whereas it is a collision-free area since it is outside the transparent cone (VO of δ-overlapping circles).

Definition III.1. ***δ-overlap:*** *Two circles each of radius r are said to δ-overlap if the length of the line joining their centers is equal to $2r - \delta$, where $0 < \delta < 2r$.*

Now, consider the standard Velocity Obstacle formulation (the basis of RVO) [36] in Figure 2 (left). From the VO formulation, we have $R_o = 2r$ and from III.1, $R_i = 2r - \delta$. Let σ denote the distance of p_j from p_i. Then from simple geometry,

$$0 \leq \Delta \leq \sin^{-1}\left(\frac{2r}{\|\sigma\|}\right) - \sin^{-1}\left(\frac{2r-\delta}{\|\sigma\|}\right)$$
$$\leq \sin^{-1}\left(\frac{2r}{\|\sigma\|}\sqrt{1-\left(\frac{\delta-2r}{\|\sigma\|}\right)^2} + \frac{\delta-2r}{\|\sigma\|}\sqrt{1-\left(\frac{2r}{\|\sigma\|}\right)^2}\right)$$

Δ is upper-bounded by a function of δ. Observe that the error bound increases for higher δ. This is interpreted as follows: the more we increase the δ-overlap, the transparent gray cone will correspondingly shrink, and the distance between the RVO generated velocity, and the ground truth velocity will increase.

We therefore require geometric representations for which $\delta \to 0$. We propose to model each pedestrian using an elliptical pedestrian representation, with the ellipse axes capturing the height of the pedestrian's visible face and the shoulder length. We do not manually differentiate between the major and minor axes since ellipses can be tall or fat depending on the pedestrian orientation, and the particular axis to map either face length or shoulder width will vary. We observe that doing so produces an effective approximation. We only use the face and the shoulder in the FRVO formulation, because occlusion makes it difficult to observe other parts of the body in dense videos reliably.

Fig. 3. (*left*) A circular representation results in many false δ-overlaps.(*right*) FRVO efficiently models pedestrians using elliptical representations that on average cause $\delta \to 0$. *Sequence:* IITF-1.

C. Computing Predicted Velocities

Each pedestrian p_i is represented using the following 8-dimensional state vector, $\Psi_t = [x, v, v_{\text{pref}}, l, w]$, where x, v , and v_{pref} denote the current position of the pedestrian's center of mass, the current velocity, and the preferred velocity, respectively. l is the height of the pedestrian's visible face, and w captures the shoulder length. v_{pref} is the velocity the pedestrian would have taken in the absence of obstacles or colliding pedestrians, computed using the standard RVO formulation.

We assume the pedestrians are oriented towards their direction of motion. For each frame and each pedestrian, we construct the half-plane constraints for each of its neighboring pedestrians and obstacles to predict its motion. We use velocity obstacles to compute the set of permitted velocities for a pedestrian. Given two pedestrians p_i and p_j, the velocity obstacle of p_i induced by p_j, denoted by $VO^\tau_{p_i|p_j}$, constitutes the set of velocities for p_i that would result in a collision with p_j at some time before τ. By definition, pedestrians p_i and p_j are guaranteed to be collision-free for at least time τ, if $\vec{v}_{p_i} - \vec{v}_{p_j} \notin VO^\tau_{p_i|p_j}$. An pedestrian p_i computes the velocity obstacle $VO^\tau_{p_i|p_j}$ for each of its neighboring pedestrians, p_j. The set of permitted velocities for a pedestrian p_i is simply the convex region given by the intersection of the half-planes of the permitted velocities induced by all the neighboring pedestrians and obstacles. We denote this convex region for pedestrian p_i and time horizon τ as $FRVO^\tau_{p_i}$. Thus, $FRVO^\tau_{p_i} = \bigcup_{p_j \in \mathcal{H}_i} VO^\tau_{p_i|p_j}$. For each pedestrian p_i, we compute the new velocity $v_{\text{new}} = \arg\max_v \|v - v_{\text{pref}}\|$ from $FRVO^\tau_{p_i}$ that minimizes the deviation from its preferred velocity v_{pref} such that $v \notin FRVO^\tau_{p_i}$.

In order to compute collision-free velocities, we need to compute the Minkowski Sums of ellipses. In practice, computing the exact Minkowski sums of ellipses is much more expensive as compared to those of circles. To overcome the complexity of exact Minkowski Sum computation, we compute conservative linear approximations of ellipses [35] and represent them as convex polygons. As a result, the collision avoidance problem reduces to linear programming. We use the predicted velocities to estimate the new position of a pedestrian in the next time-step.

IV. DensePeds and Sparse Features

In this section, we describe our tracking algorithm, DensePeds, which is based on the tracking-by-detection

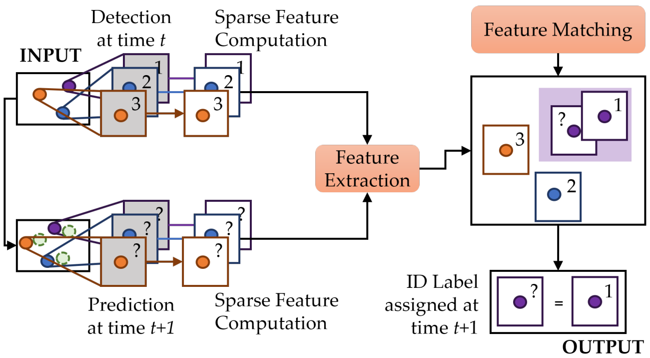

Fig. 4. Overview of our real-time pedestrian tracking algorithm, DensePeds. An input frame at time t undergoes pedestrian detection to generate segmented boxes using Mask R-CNN to compute sparse features. FRVO predicts pedestrian states for frame $t+1$ for which we also compute sparse feature. We extract features using a convolutional neural network, that are then matched using association algorithms. The predicted state receives a new ID depending on the detected person with which it is matched.

paradigm. Then, we present our approach for sparse feature extraction and show how they reduce the number of false negatives during tracking.

A. DensePeds: Pedestrian Tracking Algorithm

Our algorithm uses the detected segmentation boxes given by Mask R-CNN to extract the sparse features. These features are matched using the Cosine metric $d(\cdot,\cdot)$ (ref. the last row of Table I) and the Hungarian algorithm [37] and are used to compute the position of each pedestrian during the next frame. We illustrate our approach in Figure 4. At current time t, given the ID labels of all pedestrians in the frame, we want to assign labels for pedestrians in the next frame. We start by using Mask R-CNN to implicitly perform pixel-wise pedestrian segmentation to generate segmented boxes, which are rectangular bounding boxes with background subtraction. Next, we predict the spatial coordinates of each person's bounding box for the next time-step using FRVO, as shown in Figure 3 and Figure 3. This results in another set of bounding boxes for each pedestrian in the next frame.

We use these sets of boxes to compute appearance feature vectors, which are matched using an association algorithm [37]. This matching process is performed in two ways: the Cosine metric and the IoU overlap [38]. The Cosine metric is used to solve the following optimization problem to identify the detected pedestrian, h_j, that is most similar to p_i.

$$h_j^* = \arg\min_{h_j}(d(f_{p_i}, f_{h_j})|p_i \in \mathcal{P}, h_j \in \mathcal{H}_i). \qquad (1)$$

The IoU overlap builds a cost matrix, ζ to measure the amount of overlap of each predicted bounding box with all nearby detection bounding box candidates. $\zeta(i,j)$ stores the IoU overlap of the bounding box of Ψ_{t+1} with that of h_j and is calculated as:

$$\zeta(i,j) = \frac{\mathbb{B}_{p_i} \cap \mathbb{B}_{h_j}}{\mathbb{B}_{p_i} \cup \mathbb{B}_{h_j}}, h_j \in \mathcal{H}_i. \qquad (2)$$

Matching a detection to a predicted measurement with maximum overlap thus becomes a max weight matching problem and we solve it efficiently using the Hungarian algorithm [37]. The ID of the pedestrian at time t is then assigned to the best matched pedestrian at time $t+1$.

B. Sparse Features

In dense crowds, the bounding boxes of nearby pedestrians have significant overlap. This overlap adds noise to the feature vector extracted from the bounding boxes. We address this problem by generating sparse feature vectors by subtracting the noisy background from the original bounding box to produce segmented boxes, which is described next.

1) Segmented Boxes Using Mask R-CNN: We use Mask R-CNN to perform pixel-wise person segmentation. In practice, Mask R-CNN essentially segments out the pedestrian from its bounding box, and thereby reduces the noise that occurs when the pedestrians are nearby.

Mask R-CNN generates a bounding box and its corresponding mask for each detected pedestrian in each frame. We create a white canvas and superimpose a pixel-wise segmented pedestrian onto the canvas using the mask. We perform detection at current time t and the output consists of bounding boxes, masks, scores, and class IDs of pedestrians.

$\mathcal{B} = \{\mathbb{B}_{h_j} \mid \mathbb{B} = [\text{top left}, m, n], h_j \in \mathcal{H}\}$ denotes the set of bounding boxes, where top left$, m, n$ denote the top left corner, width, height of $\mathbb{B}_j$, respectively.

$\mathcal{M} = \{\mathbb{M}_{h_j} \mid h_j \in \mathcal{H}\}$ denotes the set of masks for each h_j, where each $\mathbb{M}_{h_j}$ is a $[m \times n]$ tensor of booleans.

Let $\mathcal{W} = \{\mathbb{W}_{h_j}(\cdot) \mid h_j \in \mathcal{H}\}$ be the set of white canvases where each canvas, $\mathbb{W}_{h_j} = [\mathbb{1}]_{m\times n}$, w and h. Then,

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

is the set of segmented boxes for each h_j at time t. These segmented boxes are used by our real-time tracking algorithm shown in Figure 4.

The segmented boxes are input to the DeepSORT CNN [19] to compute binary feature vectors. Since a large portion of a segmented box contains zeros, these feature vectors are mostly sparse. We refer to these vectors as our sparse feature vectors. Our choice of the network parameters governs the size of the feature vectors.

2) Reduced Probability of Track Loss: We now show how sparse feature vectors generated from segmented boxes reduce the probability of the loss of pedestrian tracks (false negatives).

We define $\mathcal{T}_t = \{\Psi_{1:t}\}$ to be the set of positively identified states for p_i until time t. We denote the time since the last update to a track ID as μ. We denote the ID of p_i as α and we represent the correct assignment of an ID to p_i as $\Gamma(\alpha)$. The threshold for the Cosine metric is $\lambda \underset{\text{i.i.d.}}{\sim} \mathbb{U}[0,1]$. The threshold for the track age, *i.e.*, the number of frames before which track is destroyed, is ξ. We denote the probability of an event that uses Mask R-CNN as the primary object detection algorithm with $\mathbb{P}^M(\cdot)$ and the probability of an event that uses a standard Faster R-CNN [13] as the primary object detection algorithm (*i.e.*, outputs bounding boxes without boundary subtraction) with $\mathbb{P}^F(\cdot)$. Finally, $\mathcal{T}_t \leftarrow \{\emptyset\}$ represents the loss of $\mathcal{T}_t$ by occlusion.

We now state and prove the following lemma.

Lemma IV.1. *For every pair of feature vectors $(f^M_{h_j}, f^F_{h_j})$ generated from a segmented box and a bounding box respectively, if $\|f^M_{h_j}\|_0 > \|f^F_{h_j}\|_0$, then $d(f_{p_i}, f^M_{h_j}) < d(f_{h_j}, f^F_{h_j})$ with probability $1 - \frac{B}{A}$, where A and B are positive integers and $A > B$.*

Fig. 5. Qualitative analysis of DensePeds on the NPLACE-2 sequence consisting of 144 pedestrians in the video. Frames are chosen with a gap of 4 seconds($\sim$ 80 frames). Each bounding box color is associated with a unique ID (displayed on the top left corner of each bounding box). **Observation:** Note the consistencies in the ID (color), for example, for pedestrians 1 (green), 85 (purple) and 9 (dark red) (at the front, from right to left).

Proof. Using the definition of the Cosine metric, the lemma reduces to proving the following,

$$f_{p_i}^T (f_{h_j}^M - f_{h_j}^F) > 0 \tag{3}$$

We pad both $f_{h_j}^M$ and $f_{h_j}^F$ such that $\|f_{h_j}^M\|_0 > ||f_{h_j}^F||_0$.

We reduce $f_{p_i}^T$, $f_{h_j}^M$, and $f_{h_j}^F$ to binary vectors, *i.e.*, vectors composed of 0s and 1s. Let $\Delta f = f_{h_j}^M - f_{h_j}^F$. We denote the number of 1s and -1s in Δf as A and B, respectively. Now, let x and y denote the L_0 norm of $f_{h_j}^M$ and $f_{h_j}^F$, respectively. From our padding procedure, we have $x > y$. Then, if $y = B$, then $x = A$ and we trivially have $A > B$. But if $y > B$, then $A = x - (y - B) \implies A - B = x - y$. From $x > y$, it again follows that $A > B$. Thus, $x > y \implies A > B$.

Next, we define a $(1, 1)$ coordinate in an ordered pair of vectors as the coordinate where both vectors contain 1s. Similarly, a $(1, -1)$ coordinate in an ordered pair of vectors is the coordinate where the first vector contains 1 and the second vector contains -1. Then, let p_a and p_b respectively denote the number of $(1, 1)$ coordinates and $(1, -1)$ coordinates in the pair $(f_{p_i}^T, \Delta f)$. By definition, we have $0 < p_a < A$ and $0 < p_b < B$. Thus, if we assume p_a and p_b to be uniformly distributed, it directly follows that $\mathbb{P}(p_a > p_b) = 1 - \frac{B}{A}$. ■

Based on Lemma IV.1, we finally prove the following proposition.

Proposition IV.1. *With probability $1 - \frac{B}{A}$, sparse feature vectors extracted from segmented boxes decrease the loss of pedestrian tracks, thereby reducing the number of false negatives in comparison to regular bounding boxes.*

Proof. In our approach, we use Mask R-CNN for pedestrian detection, which outputs bounding boxes and their corresponding masks. We use the mask and bounding box pair to generate a segmented box (Section IV-B.1). The correct assignment of an ID depends on successful feature matching between the predicted measurement feature and the optimal detection feature, that is, when the cosine cost between the two features is below a set threshold, λ. But since the correct ID assignment depends on multiple factors including a low cosine cost, we instead exploit the equivalency of the contrapositive,

$$d(f_{p_i}, f_{h_j^*}) > \lambda \Leftrightarrow (\alpha = \emptyset) \tag{4}$$

Equation 4 indicates that when the cosine cost is greater than λ, then feature matching fails and therefore an ID fails to be assigned, or equivalently, set to $\emptyset$. Using Lemma IV.1 and the fact that $\lambda \underset{i.i.d.}{\sim} \mathbb{U}[0, 1]$,

$$\mathbb{P}(d(f_{p_i}, f_{h_{j,p_i}^*}^M) > \lambda) < \mathbb{P}(d(f_{p_i}, f_{h_{j,p_i}^*}^F) > \lambda)$$

Using the equivalency in Eq. 4, we obtain,

$$\mathbb{P}^M(\alpha = \emptyset) < \mathbb{P}^F(\alpha = \emptyset) \tag{5}$$

Now, in our approach, we set certain fixed conditions that need to be satisfied for a track to be destroyed or lost. Those conditions are listed as follows:

1) DensePeds updates the ID of every pedestrian after each frame. If an update fails to occur for $\mu > \zeta$ frames, then this leads to loss of the track of that pedestrian.
2) If the first condition is satisfied, and the current ID of a pedestrian is not set, then the track of that pedestrian is lost.

We formalize the two conditions as follows,

$$(\mu > \xi) \wedge (\alpha = \emptyset) \Leftrightarrow \mathcal{T}_t \leftarrow \{\emptyset\}$$

Using Eq. 5, it follows that,

$$\mathbb{P}^M(\mathcal{T}_t \leftarrow \{\emptyset\}) < \mathbb{P}^F(\mathcal{T}_t \leftarrow \{\emptyset\}) \tag{6}$$

Informally, equation 6 tells us that the probability of losing a track using segmented boxes is less than the probability of losing a track if we were to use regular bounding boxes.

To complete the proof, we now show that equation 6 implies that fewer lost tracks leads to fewer false negatives. We start by defining the total number of false negatives (FN) as

$$FN = \sum_{t=1}^{T} \sum_{p_g \in \mathcal{G}} \delta_{\mathcal{T}_t} \tag{7}$$

where $p_g \in \mathcal{G}$ denotes a ground truth pedestrian in the set of all ground truth pedestrians at current time t and $\delta_z = 1$ for $z = 0$ and 0 elsewhere. This is a variation of the Kronecker delta function. Using Eq. 6 and Eq. 7, we can say that fewer lost tracks ($\mathcal{T}_t \leftarrow \{\emptyset\}$) indicate a smaller number of false negatives. We empirically demonstrate this analysis in Section V. ■

The upper bound, $\mathbb{P}^F(\mathcal{T}_t)$, in Eq. 6 depends on the amount of padding done to f_{p_i} and f_{h_j}. A general observed trend is that a higher amount of padding results in a larger upper bound in Eq. 6.

V. Experimental Evaluation

A. Datasets

We evaluate DensePeds on both our dense pedestrian crowd dataset and on the popular MOT [8] benchmark. MOT is a standard benchmark for testing the performance of multiple object tracking algorithms, and it subsumes older benchmarks such as KITTI [46] and PETS [47]. When evaluating on MOT, we use the popular MOT16 and MOT15 sequences. However, the main drawback of the sequences in the MOT benchmark is that they do not contain dense crowd videos that we address in this paper.

Our dataset consists of 8 dense crowd videos where the crowd density ranges from 2 to 2.7 pedestrians per square meter. By comparison, the densest crowd video in the MOT benchmark has around 0.2 persons per square meter. All videos in our dataset are shot at an elevated view using a stationary camera. Seven of the eight videos are shot 1080p resolution, and one video is at 480p resolution.

While we agree that the current size of our dataset is prohibitively small for training existing deep learning-based object detectors, we showcase the results of our method on this dataset to validate our claims and analyses. We plan to expand and release a benchmark version of our dataset in the recent future.

B. Evaluation Metrics and Methods

1) Metrics: We use a subset of the standard CLEAR MOT metrics [48] for evaluating the performance of our algorithm. Specifically, the metrics we use are:

1) **Number of mostly tracked trajectories (MT).** Correct ID association with a pedestrian across frames for at least 80% of its life span in a video.
2) **Number of mostly lost trajectories (ML).** Correct ID association with a pedestrian across frames for at most 20% of its life span in the video.
3) **False Negatives (FN).** Total number of false negatives in object detection (*i.e.*, pedestrian tracks lost) over the video.
4) **Identity Switches (IDSW).** The total number of identity switches over pedestrians in the video.
5) **MOT Precision (MOTP).** The percentage misalignment between all the predicted bounding boxes and corresponding ground truth boxes over the video.
6) **MOT Accuracy (MOTA).** Overall tracking accuracy taking into account false positives (FP), false negatives (FN), and ID switches (IDSW). It is defined as $MOTA = 1 - \frac{FP + FN + IDS}{GT}$, where GT is the sum of annotated pedestrians in all the frames in the video.

These are the most important metrics in CLEAR MOT for evaluating a tracker's performance. We exclude metrics that evaluate the performance of detection since object detection is not a contribution of this paper. We have also excluded the number of false positives from our tables since it is already included in the definition of MOTA and is not the focus of this paper. On the other hand, we highlight the results of false negatives to validate the theoretical formulation of our algorithm in Section III-B.

2) Methods: There is a large number of tracking-by-detection methods on the MOT benchmark, and it is not feasible to evaluate our method against all of them in the limited scope of this paper. Further, many methods submitted on the MOT benchmark server are unpublished. Thus, to keep our evaluations as fair and competitive as possible, we compare with state-of-the-art online trackers on the MOT16 as well as MOT15 benchmarks that are published. By "state-of-the-art", we refer to methods that have an average rank greater or equal to our average rank.

For evaluation on our dataset, in addition to online state-of-the-art published methods, we also require their code. The only methods that meet all these criteria are MOTDT [39] and MDP [40].

C. Results

All results generated by DensePeds are obtained without any training on tracking datasets. Thus, DensePeds does not suffer from over-fitting or dataset bias. All the methods that we compare with are trained on the MOT sequences.

1) Our Dataset: Table II compares DensePeds with MOTDT [39] and MDP [40] on our dense crowd dataset. MOTDT is currently the best online published tracker on the MOT16 benchmark with available code, and MDP is the second-best method with available code. We used their off-the-shelf implementations, with weights pre-trained on the MOT benchmark, to compare with DensePeds.

We observe that DensePeds produces the lowest false negatives of all the methods. Thus, by Proposition IV.1 and Table II, DensePeds provably reduces the number of false negatives on dense crowd videos.

The low MT and high ML scores for the sequences are not an indicator of poor performance; rather they are a consequence of the strict definitions of the metrics. For example, in dense crowd videos, it is challenging to maintain a track for 80% of a pedestrian's life span. DensePeds has the highest MT and lowest ML percentages of all the methods. This explains our high number of ID switches as a higher number of pedestrians being tracked would correspondingly increase the likelihood of ID switches. Most importantly, the MOTA of DensePeds is 2.9% more than that of MOTDT and 2.6% more than that of MDP. Roughly speaking, this is equivalent to an average rank difference of 19 from MOTDT and 17 from MDP on the MOT benchmark. We do not include the tracking speed metric (Hz) as shown in Table III since that is a metric produced exclusively by the MOT benchmark server.

2) MOT Benchmark: Although our algorithm is primarily targeted for dense crowds, in the interest of thorough analysis, we also evaluate DensePeds on sparser crowds. We select a wide range of methods (as described in Section V-B.2) to compare with DensePeds and highlight its relative advantages and disadvantages in Table III.

As expected, we do not outperform state of the art on the MOT sequences due to its sparse crowd sequences. We specifically point to our low MOTA scores which we attribute to our algorithm requiring a more accurate ground truth than the ones provided by the MOT benchmark. Our detection and tracking are highly sensitive — they can track pedestrians that are too distant to be manually labeled. This erroneously leads to a higher count of false positives and reduces the MOTA. This was observed to be true for the methods we

TABLE II

EVALUATION ON OUR DENSE CROWDS DATASET WITH MOTDT [39] AND MDP [40]. MOTDT IS CURRENTLY THE BEST *online* TRACKER ON THE MOT BENCHMARK WITH OPEN-SOURCED CODE. BOLD IS BEST. ARROWS (↑, ↓) INDICATE THE DIRECTION OF BETTER PERFORMANCE. **OBSERVATION:** DENSEPEDS IMPROVES THE ACCURACY (MOTA) OVER THE STATE-OF-THE-ART BY 2.6%. WE REDUCE THE NUMBER OF FALSE NEGATIVES (FN) BY 11% COMPARED TO THE NEXT BEST METHOD, WHICH IS A DIRECT CONSEQUENCE OF THE THEORETICAL FORMULATION IN SECTION III-B.

Sequence Name	Tracker	MT(%)↑	ML(%)↓	IDS↓	FN↓	MOTP(%)↑	MOTA(%)↑
IITF-1	MOTDT	**1.8**	81.1	84 (0.2%)	10,050 (29.3%)	**61.2**	70.5
	MDP	**1.8**	81.1	**40 (0.1%)**	10,287 (30.0%)	**61.2**	69.9
	DensePeds	**1.8**	**43.4**	194 (0.5%)	**8,357 (24.3%)**	60.7	**75.1**
IITF-2	MOTDT	5.7	40.0	51 (0.3%)	3,902 (23.6%)	70.9	76.1
	MDP	14.2	25.7	**36 (0.2%)**	3,033 (18.3%)	**72.6**	81.4
	DensePeds	**17.1**	**11.5**	82 (0.5%)	**2,308 (14.0%)**	70.3	**85.5**
IITF-3	MOTDT	2.0	12.5	187 (0.5%)	7,563 (19.4%)	69.5	80.1
	MDP	8.3	16.6	**85 (0.2%)**	7,735 (19.8%)	**71.0**	79.9
	DensePeds	**22.9**	**4.2**	204 (0.5%)	**5,968 (15.3%)**	69.0	**84.2**
IITF-4	MOTDT	11.4	17.2	83 (0.5%)	3,025 (19.2%)	68.2	80.2
	MDP	**14.2**	28.5	**41 (0.3%)**	3,129 (19.9%)	**70.3**	79.8
	DensePeds	**14.2**	**5.8**	114 (0.7%)	**2,359 (15.0%)**	67.6	**84.3**
NDLS-1	MOTDT	0	78.3	113 (0.3%)	9,965 (28.9%)	61.8	70.8
	MDP	0	89.1	**72 (0.2%)**	10,293 (29.8%)	**62.0**	70.0
	DensePeds	0	**76.1**	131 (0.4%)	**9,728 (28.2%)**	61.5	**71.4**
NDLS-2	MOTDT	0	85.2	142 (0.3%)	14,084 (28.9%)	63.0	70.8
	MDP	0	85.1	**82 (0.2%)**	13,936 (28.6%)	**63.9**	71.2
	DensePeds	0	**74.5**	183 (0.4%)	**13,240 (27.2%)**	62.7	**72.5**
NPLACE-1	MOTDT	**1.9**	64.7	181 (0.5%)	9,676 (27.7%)	**63.3**	71.8
	MDP	**1.9**	64.7	**153 (0.4%)**	9,629 (27.5%)	63.1	72.0
	DensePeds	**1.9**	**51.0**	210 (0.6%)	**9,114 (26.1%)**	62.8	**73.3**
NPLACE-2	MOTDT	**3.8**	67.3	83 (0.3%)	7,443 (27.4%)	**63.2**	72.3
	MDP	5.7	69.2	**54 (0.2%)**	7,484 (27.6%)	62.8	72.2
	DensePeds	**3.8**	**61.6**	90 (0.3%)	**7,161 (26.4%)**	62.8	**73.3**
Summary	MOTDT	2.9	58.0	924 (0.4%)	65,708 (26.2%)	66.1	73.4
	MDP	5.1	59.9	**563 (0.2%)**	65,526 (26.1%)	**67.4**	73.7
	DensePeds	**7.0**	**43.4**	1,208 (0.5%)	**58,235 (23.2%)**	65.7	**76.3**

TABLE III

EVALUATION ON THE MOT BENCHMARK WITH NON-ANONYMOUS, ONLINE METHODS. IN THE INTEREST OF A FAIR COMPARISON, WE HAVE CHOSEN METHODS THAT HAVE AN AVERAGE RANK EITHER GREATER THAN OR EQUAL TO OUR AVERAGE RANK. BOLD IS BEST. ARROWS (↑, ↓) INDICATE THE DIRECTION OF BETTER PERFORMANCE. **OBSERVATION:** DENSEPEDS IS UP TO 4.5× FASTER THAN PREVIOUS METHODS AND IMPROVES ACCURACY BY UP TO 17%. ITS OVERALL PERFORMANCE HAS A RANK OF 23.5 (TOP 13%) OF ALL PUBLISHED METHODS ON THE MOT15 AND 28.8 (TOP 20%) ON MOT16.

	Tracker	Hz↑	MT(%)↑	ML(%)↓	IDS↓	FN↓	MOTP(%)↑	MOTA(%)↑
MOT15	AMIR15 [41]	1.9	15.8	**26.8**	1026	29,397	71.7	37.6
	HybridDAT [42]	4.6	11.4	42.2	358	31,140	72.6	35.0
	AM [17]	0.5	11.4	43.4	**348**	34,848	70.5	34.3
	AP_HWDPL_p [43]	6.7	8.7	37.4	586	33,203	72.6	**38.5**
	DensePeds	**28.9**	**18.6**	32.7	429	**27,499**	**75.6**	20.0
MOT16	EAMTT_pub [44]	11.8	7.9	49.1	965	102,452	75.1	38.8
	RAR16pub [45]	0.9	13.2	41.9	648	91,173	74.8	45.9
	STAM16 [17]	0.2	14.6	43.6	**473**	91,117	74.9	46.0
	MOTDT [39]	**20.6**	15.2	38.3	792	85,431	74.8	**47.6**
	AMIR [41]	1.0	14.0	41.6	774	92,856	**75.8**	47.2
	DensePeds	18.8	**20.3**	**36.1**	722	**78,413**	75.5	40.9

compared with as well. Consequently, we exclude FP from the calculation of MOTA for all methods in the interest of fair evaluation.

However, following Proposition IV.1, our method achieves the lowest number of false negatives among all the methods on the MOT benchmark as well. In terms of runtime, we are approximately 4.5× faster than the state-of-the-art methods on an NVIDIA Titan Xp GPU.

3) Ablation Experiments: In Table IV, we validate the benefits of sparse feature vectors (obtained via segmented boxes) in detection and FRVO in tracking in dense crowds with ablation experiments.

First, we replace segmented boxes with regular bounding boxes (boxes without background subtraction), and compare its performance with DensePeds on the MOT benchmark. In accordance with Proposition IV.1, DensePeds reduces the number of false negatives by 20.7% on relative.

Next, we replace FRVO in turn with a constant velocity motion model [19], the Social Forces model [31], and the standard RVO model [6]; all without changing the rest of the DensePeds algorithm. All of these models fail to track pedestrians in dense crowds (ML nearly 100%). Note that a consequence of failing to track pedestrians is the unusually low number of ID switches that we observe for these methods. This, in turn, leads to their comparatively high MOTA, despite being mostly unable to track pedestrians. Using FRVO, we decrease ML by 55% on relative. At the same time, our MOTA improves by a significant 12% on relative and 8.5% on absolute.

VI. ACKNOWLEDGEMENT

This research is supported in part by ARO grant W911NF19-1- 0069, Alibaba Innovative Research (AIR) program and Intel.

REFERENCES

[1] A. Bera, T. Randhavane, and D. Manocha, “Aggressive, tense or shy? identifying personality traits from crowd videos,” in *IJCAI*, 2017.

[2] R. Chandra, U. Bhattacharya, A. Bera, and D. Manocha, “Traphic: Trajectory prediction in dense and heterogeneous traffic using weighted interactions,” in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019.

[3] A. Teichman and S. Thrun, “Practical object recognition in autonomous driving and beyond,” in *Advanced Robotics and its Social Impacts*, pp. 35–38, IEEE, 2011.

[4] L. Zhang and L. van der Maaten, “Structure preserving object tracking,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 1838–1845, 2013.

TABLE IV

ABLATION STUDIES WHERE WE DEMONSTRATE THE ADVANTAGE OF SEGMENTED BOXES AND FRVO. WE REPLACE SEGMENTED BOXES WITH REGULAR BOUNDING BOXES (BBOX) AND COMPARE ITS PERFORMANCE WITH DENSEPEDS ON THE MOT BENCHMARK. WE ALSO REPLACE THE FRVO MODEL IN TURN WITH A CONSTANT VELOCITY MODEL (CONST. VEL), THE SOCIAL FORCES MODEL (SF) [31] AND THE STANDARD RVO MODEL (RVO) [6] AND COMPARE THEM ON OUR DENSE CROWD DATASET. BOLD IS BEST. ARROWS ($\uparrow, \downarrow$) INDICATE THE DIRECTION OF BETTER PERFORMANCE. **OBSERVATION:** USING FRVO IMPROVES THE MOTA BY 13.3% OVER THE NEXT BEST ALTERNATE. USING SEGMENTED BOXES REDUCES THE FALSE NEGATIVES BY 20.7%.

Detection	MT(%)↑	ML(%)↓	IDS↓	FN↓	MOTP(%)↑	MOTA(%)↑
BBox	14.0	44.7	**313**	34,716	**76.4**	17.6
DensePeds	**19.0**	**32.7**	429	**27,499**	75.6	**20.0**
Motion Model	MT(%)↑	ML(%)↓	IDS↓	FN↓	MOTP(%)↑	MOTA(%)↑
Const. Vel	0	98.0	33	82,467	64.2	67.1
SF	0	96.7	145	81,936	64.5	67.3
RVO	0	96.1	155	81,832	64.4	67.8
DensePeds	**7.0**	**43.4**	1,208	**58,235**	**65.7**	**76.3**

[5] W. Hu, X. Li, W. Luo, X. Zhang, S. Maybank, and Z. Zhang, "Single and multiple object tracking using log-euclidean riemannian subspace and block-division appearance model," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 34, no. 12, pp. 2420–2440, 2012.

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

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

[8] A. Milan, L. Leal-Taixe, I. Reid, S. Roth, and K. Schindler, "MOT16: A Benchmark for Multi-Object Tracking," *ArXiv e-prints*, Mar. 2016.

[9] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," in *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*, vol. 1, pp. 886–893, IEEE, 2005.

[10] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *International journal of computer vision*, vol. 60, no. 2, pp. 91–110, 2004.

[11] R. Girshick, J. Donahue, T. Darrell, and J. Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation," *ArXiv e-prints*, Nov. 2013.

[12] R. Girshick, "Fast r-cnn," in *Proceedings of the IEEE international conference on computer vision*, pp. 1440–1448, 2015.

[13] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks," *ArXiv e-prints*, June 2015.

[14] R. Li, X. Dong, Z. Cai, D. Yang, H. Huang, S.-H. Zhang, P. Rosin, and S.-M. Hu, "Pose2seg: Human instance segmentation without detection," *arXiv preprint arXiv:1803.10683*, 2018.

[15] A. Milan, S. H. Rezatofighi, A. Dick, I. Reid, and K. Schindler, "Online multi-target tracking using recurrent neural networks," in *Thirty-First AAAI Conference on Artificial Intelligence*, 2017.

[16] S.-H. Bae and K.-J. Yoon, "Confidence-based data association and discriminative deep appearance learning for robust online multi-object tracking," *IEEE transactions on pattern analysis and machine intelligence*, vol. 40, no. 3, pp. 595–610, 2018.

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

[18] K. Fang, Y. Xiang, and S. Savarese, "Recurrent autoregressive networks for online multi-object tracking," *arXiv preprint arXiv:1711.02741*, 2017.

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

[20] S. Murray, "Real-time multiple object tracking-a study on the importance of speed," *arXiv preprint arXiv:1709.03572*, 2017.

[21] A. Milan, S. Roth, and K. Schindler, "Continuous energy minimization for multitarget tracking," *IEEE transactions on pattern analysis and machine intelligence*, vol. 36, no. 1, pp. 58–72, 2013.

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

[23] R. Henschel, L. Leal-Taixe, D. Cremers, and B. Rosenhahn, "Fusion of head and full-body detectors for multi-object tracking," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pp. 1428–1437, 2018.

[24] 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*, pp. 233–243, Springer, 2017.

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

[26] A. Bera and D. Manocha, "Realtime multilevel crowd tracking using reciprocal velocity obstacles," in *Pattern Recognition (ICPR), 2014 22nd International Conference on*, pp. 4164–4169, IEEE, 2014.

[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*, pp. 1801–1808, IEEE, 2014.

[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*, pp. 261–268, Sept 2009.

[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*, pp. 1345–1352, IEEE, 2011.

[30] C. W. Reynolds, "Flocks, herds and schools: A distributed behavioral model," in *ACM SIGGRAPH computer graphics*, vol. 21, pp. 25–34, ACM, 1987.

[31] D. Helbing and P. Molnar, "Social force model for pedestrian dynamics," *Physical review E*, vol. 51, no. 5, p. 4282, 1995.

[32] I. Karamouzas, P. Heil, P. Van Beek, and M. H. Overmars, "A predictive collision avoidance model for pedestrian simulation," in *International Workshop on Motion in Games*, pp. 41–52, Springer, 2009.

[33] G. Antonini, S. V. Martinez, M. Bierlaire, and J. P. Thiran, "Behavioral priors for detection and tracking of pedestrians in video sequences," *International Journal of Computer Vision*, vol. 69, no. 2, pp. 159–180, 2006.

[34] S.-Y. Chung and H.-P. Huang, "A mobile robot that understands pedestrian spatial behaviors," in *Intelligent Robots and Systems (IROS), 2010 IEEE/RSJ International Conference on*, pp. 5861–5866, IEEE, 2010.

[35] A. Best, S. Narang, and D. Manocha, "Real-time reciprocal collision avoidance with elliptical agents," in *Robotics and Automation (ICRA), 2016 IEEE International Conference on*, pp. 298–305, IEEE, 2016.

[36] P. Fiorini and Z. Shiller, "Motion planning in dynamic environments using velocity obstacles," *The International Journal of Robotics Research*, vol. 17, no. 7, pp. 760–772, 1998.

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

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

[39] C. Long, A. Haizhou, Z. Zijie, and S. Chong, "Real-time multiple people tracking with deeply learned candidate selection and person re-identification," in *ICME*, 2018.

[40] Y. Xiang, A. Alahi, and S. Savarese, "Learning to track: Online multi-object tracking by decision making," in *Proceedings of the IEEE international conference on computer vision*, pp. 4705–4713, 2015.

[41] A. Sadeghian, A. Alahi, and S. Savarese, "Tracking the untrackable: Learning to track multiple cues with long-term dependencies,"

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

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

[44] R. Sanchez-Matilla, F. Poiesi, and A. Cavallaro, "Online multi-target tracking with strong and weak detections," in *European Conference on Computer Vision*, pp. 84–99, Springer, 2016.

[45] K. Fang, Y. Xiang, and S. Savarese, "Recurrent autoregressive networks for online multi-object tracking," *2018 IEEE Winter Conference on Applications of Computer Vision (WACV)*, pp. 466–475, 2018.

[46] A. Geiger, P. Lenz, C. Stiller, and R. Urtasun, "Vision meets robotics: The kitti dataset," *The International Journal of Robotics Research*, vol. 32, no. 11, pp. 1231–1237, 2013.

[47] L. Patino, T. Cane, A. Vallee, and J. Ferryman, "Pets 2016: Dataset and challenge," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pp. 1–8, 2016.

[48] K. Bernardin and R. Stiefelhagen, "Evaluating multiple object tracking performance: the clear mot metrics," *Journal on Image and Video Processing*, vol. 2008, p. 1, 2008.