REGROUP: A Robot-Centric Group Detection and Tracking System

Angelique Taylor
Computer Science & Engineering
University of California, San Diego
amt062@eng.ucsd.edu

Laurel D. Riek
Computer Science & Engineering
University of California, San Diego
lriek@eng.ucsd.edu

***Abstract*—To facilitate HRI's transition from dyadic to group interaction, new methods are needed for robots to sense and understand team behavior. We introduce the Robot-Centric Group Detection and Tracking System (REGROUP), a new method that enables robots to detect and track groups of people from an ego-centric perspective using a crowd-aware, tracking-by-detection approach. Our system employs a novel technique that leverages person re-identification deep learning features to address the group data association problem. REGROUP is robust to real-world vision challenges such as occlusion, camera egomotion, shadow, and varying lighting illuminations. Also, it runs in real-time on real-world data. We show that REGROUP outperformed three group detection methods by up to 40% in terms of precision and up to 18% in terms of recall. Also, we show that REGROUP's group tracking method outperformed three state-of-the-art methods by up to 66% in terms of tracking accuracy and 20% in terms of tracking precision. We plan to publicly release our system to support HRI teaming research and development. We hope this work will enable the development of robots that can more effectively locate and perceive their teammates, particularly in uncertain, unstructured environments.**

***Index Terms*—human robot interaction, group detection, group tracking, social robot navigation, deep learning**

I. Introduction

Increasingly, people expect robots to interact fluently with them in crowded, real-world settings. For example, assisting families in public places (e.g., airports and hotels [1], [2], assisting clinical teams [3], [4] or transporting people via autonomous vehicles [5], [6], [7], [8]. In these real-world settings, which have considerable uncertainty, robots need robust perception methods to accomplish their tasks safely and effectively [9], [10], [11], [12]. One key feature of these environments is that people often interact in groups, a fact which robots can leverage to interact more fluently with human teammates [13], [14], [15], [16], [17].

The field of Human-Robot Interaction (HRI) has recognized the importance of transitioning from dyadic, lab-based interactions to group-based, real-world settings teams [18], [19], [20], [21], [22]. Thus, we need new methods to support this transition. This motivates us to focus on group perception methods because they can be used to address critical problems that robots encounter in real-world group settings, for example, when robots can potentially harm people around them as a result of delays or misclassifications in their perception systems [23], [24], [25].

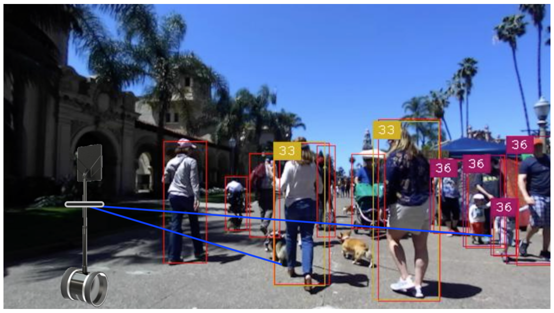

Fig. 1: REGROUP enables robots to detect and track groups of people using a crowd-aware, tracking-by-detection approach.

Prior work in vision and robotics has explored group detection from exocentric and ego-centric sensing perspectives [26]. Methods that employ exocentric sensing rely on stationary, overhead cameras [27], [28]. These methods represent pedestrians as points on the ground plane to build models that learn trajectory patterns. They then employ probabilistic methods [29], [30], [31], [32], graph-based approaches [33], [34], [35], or social force models [36], [37], [38] to detect groups. However, these methods are less helpful for real-world HRI group applications, as they require placing external sensors in the environment.

Instead, methods that rely on ego-centric sensors tend to be more useful. Here, sensors are place on a robot (or person), and various methods that generate pedestrian detections [39], [40], [41]. However, most prior work does not consider the social dynamics in the environment; doing so could enable more socially-aware navigation. A key difference ego-centric approaches is that they employ ego-centric image feature extraction techniques to model a pedestrian's appearance, which is particularly useful for mobile robotic applications.

There are several common ego-centric perception methods, including probabilistic methods such as Multiple Hypothesis Tracking (MHT) [42], [43], [44], fluid dynamics-inspired models [45], and clustering [46], [47]. However, we focus on commonly used methods such as MHT and clustering.

One example of an HRI system that uses MHT for ego-

This work was partially supported by the National Science Foundation under Grant Nos. IIS-1734482 and DGE-1650112, and the Microsoft Research Dissertation Award. We also thank Darren Chan, Wesley Xiao, Ryan Chu, and Per Antoine Carlsen for supporting data collection and annotation efforts.

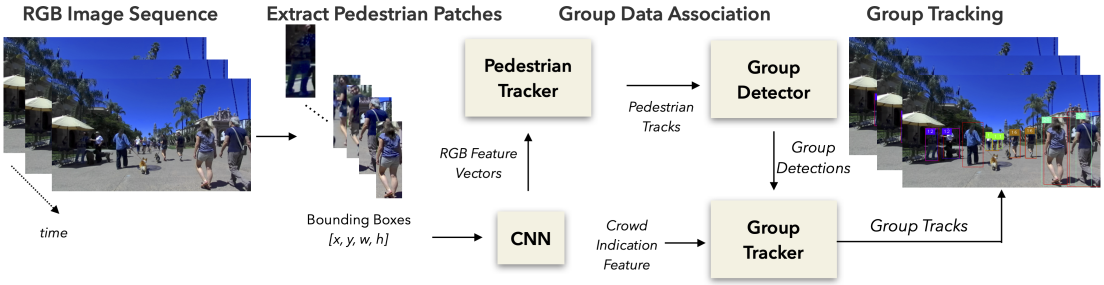

Fig. 2: Given video data, REGROUP extracts pedestrian patches and extracts appearance descriptors from them using a CNN. Then, it uses these descriptors to track pedestrians. REGROUP detects detector using the pedestrian tracks followed by tracking groups using our novel crowd indication feature (CIF), which enables REGROUP to handle high levels of occlusion.

centric perception is Spencer, which is also a group detection and tracking system [1]. This project focused on designing an assistive robotic platform that guides people in busy airports. Under this project, [40] published a framework that employs various sensor fusion, pedestrian detection, and group tracking methods. Although this work made great progress to address the ego-centric group detection and tracking problem, its main drawback is that it required manually-annotated training data for group detection, data association, and social groupings. Additionally, MHT is computationally expensive because it uses tree-based data structures that grow exponentially with the number of pedestrians. Other methods used clustering to predict moving pedestrians in crowds by estimating clusters in 3D point clouds [46]. However, the authors employed a static sensor setup; this method may fail when employed on mobile robot as mobility can increase noise in the point cloud.

There are three major limitations in prior group detection methods which require further investigation. First, most prior work relies on stationary, exocentric, overhead sensor setups which cannot be accessed by mobile robots working in new environments [48], [31], [30]. Second, many existing techniques require *a priori* knowledge of groups, and need to be trained from large datasets that must be manually annotated.Third, public spaces are often crowded, which causes error propagation over time in modern tracking systems (for pedestrians and groups) and leads to degrading performance.

The goal of the data association problem is to match objects from one timestep to the next. Many methods employ Convolutional Neural Network (CNN) appearance descriptors for data association between pedestrians [49], [50], [51]. Person re-identification CNNs are commonly used to generate such appearance descriptors as they are invariant to changes in scale, rotation, and lighting [52], [53]. However, there is a lack of work that explores this approach for tracking groups.

To address these gaps, we introduce the *Robot-Centric Group Detection and Tracking System* (REGROUP), an ego-centric group detection and tracking system (See Figure 1). Inspired by Robot-Centric Group Estimation Model (RoboGEM) [47], we aim to improve upon this prior work by designing a new group detection system and developing a new group tracking system. REGROUP uses a tracking-by-detection approach with a person re-identification CNN for group data association. We employ motion and appearance distance metrics to track group states over time. Additionally, we propose a effective technique that detects when the environment is crowded, to enable REGROUP to handle high levels of occlusion in real-world environments. Furthermore, REGROUP runs at 45.3 frames-per-second on a real-world dataset.

The **contributions of this paper** are as follows:

1) We introduce a new ego-centric group detection and tracking system using a crowd-aware, tracking-by-detection technique. Our system leverages person re-identification deep learning feature activation maps to address the group data association problem. We show that REGROUP is robust to real-world vision challenges such as occlusion, camera egomotion, and shadow.
2) We show that it runs in real-time on real-world data.
3) We show that REGROUP outperforms three state-of-the-art group detection and tracking methods [54], [55], [56].
4) We plan to publicly release our system to enable robotics researchers to design intelligent systems for group HRI.

Our work addresses the problem of group detection and tracking, which is essential for robots to effectively team with multiple collocated people. Our work also propels exploration in the broader robotics community to advance research in areas including autonomous vehicles and multi-robot systems.

II. REGROUP

REGROUP is an ego-centric group detection and tracking system that runs on RGB video data, in real-time. It runs online using an RGB camera or offline with pedestrian detections precomputed and stored in memory. We define groups as people spatially close to each other with a common motion goal [40]. We capture this intuition in our group detection algorithm using three distance metrics designed particularly for ego-centric perception. Also, we present a crowd indication feature (CIF) which enables robots to track in crowded environments.

Figure 2 shows an overview of REGROUP. Starting with a RGB video, REGROUP detects pedestrians, extracts their pedestrian patches, and passes those patches to a Convolutional Neural Network (CNN). The CNN generates an appearance descriptor which is used for data association of pedestrians and groups. Then, REGROUP uses these appearance descriptors to track pedestrians. Next, it detects groups using our distance metrics (See Figure 3). Finally, it performs our new group data association technique using fused group appearance

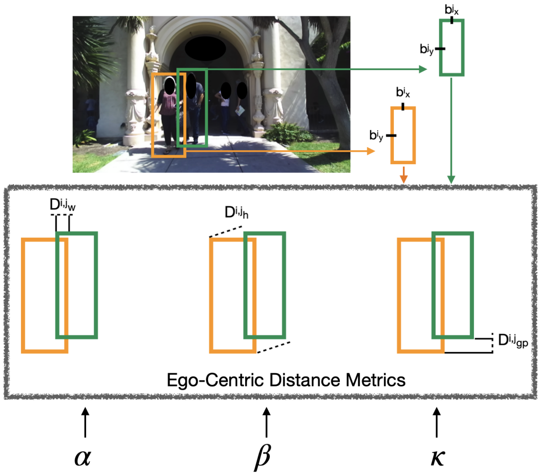

Fig. 3: REGROUP's group detection distance metrics.

descriptors, and tracks groups using Kalman filtering. In this section, we discuss each of these steps in detail.

A. Ego-centric Group Detection

We introduce a new ego-centric group detection algorithm that has three main steps. First, it employs a state-of-the-art pedestrian detection method [57] that achieves real-time performance on a GPU. This method outputs bounding boxes (BB) that are parameterized by $\langle x, y, w, h \rangle$ which are the center column, center row, width, and height respectively. Second, we employ the pedestrian tracking method by [58] that achieves real-time performance. Algorithm 1 shows our ego-centric group detection method. Consider a scenario in which pedestrian i, p^i and pedestrian j, p^j, have bounding boxes b^i and b^j $(i \neq j)$, respectively.

We conducted iterative experiments to explore distance metrics for the ego-centric group detection problem. We started with the metrics employed in RoboGEM [47]. Then, we expanded them to explore other distance metrics using the pedestrian BBs. We made several observations such as pedestrians that are nearby tend to have larger visual representations than people far away. We found that the combination of "inner distance" between pedestrians, the distance between pedestrians' lower body (e.g., indicating the ground plane), and the ratio of the height of two adjacent pedestrians generated the best performance of all approaches we tested.

We generate three $N_t \times N_t$ adjacency matrices (AM) for the width D_w between b^i and b^j, height ratio D_h between b^i and b^j, and ground plane distance D_{gp} between b^i and b^j, where N_t is the number of pedestrians at time t (See Figure 3). The adjacency matrix captures different distance metrics between pedestrians in the scene. We define D_w as the inner distance between b^i and b^j which groups pedestrians with a small space between them. D_h measures how close the height of b^i matches b^j which groups pedestrians that are close to the robot together and it groups those that are far away from the robot together. Finally, D_{gp} measures the ground plane distance between b^i and b^j which groups pedestrians based on how close they are walking near each other.

Algorithm 1: Ego-centric Group Detection

```
Input  : Pedestrian Detection BB = $\{b^n | n = 1, \dots, N_t\}$
         Height and Ground Plane Thresholds $\alpha$ and $\kappa$
Output: Group detection IDs $G = \{1, \dots, N_t\}$
$D = D_w = D_h = D_{gp} = \{\}$ // adjacency matrices of width,
 height ratio, and ground plane distance between pedestrians.
for $i \in \{1, \dots, N_t\}$ do
    for $j \in \{1, \dots, N_t\}$ do
        if $i \neq j$ then
            $\mathrm{D}_w^{i,j} = \begin{cases} |b_x^i - b_x^j| - \frac{|b_w^i + b_w^j|}{2}, & \text{if } D_w^{i,j} \leq \beta \\ 0, & \text{otherwise} \end{cases}$
            $\mathrm{D}_h^{i,j} = \begin{cases} \frac{min(b_h^i, b_h^j)}{max(b_h^i, b_h^j)}, & \text{if } D_h^{i,j} \leq \alpha \\ 0, & \text{otherwise} \end{cases}$
            $\mathrm{D}_{gp}^{i,j} = \begin{cases} |(b_y^i - b_h^i) - (b_y^j - b_h^j)|, & \text{if } D_{gp}^{i,j} \leq \kappa \\ 0, & \text{otherwise} \end{cases}$
            _
$D = D_w \circ D_h \circ D_{gp}$
$G \leftarrow$ DFS($D$)
Return $G$
```

The AMs capture whether p^i and p^j are in the same group. We apply numerical thresholds β, α, and κ to D_w, D_h, and D_{gp} respectively such that we exclude group candidates that clusters pedestrians that are not physically close to each other. β is the mean of b_w^i and b_w^j. $\alpha \in [0, 1]$ is the height ratio between b_h^i and b_h^j. $\kappa \in [1, I_h]$ is the ground plane distance (in pixels) between b^i and b^j where I_h is the image height.

Next, we normalize D_w, D_h, and D_{gp}; therefore, a value of 1 would indicate a group candidate and a value of 0 means that b^i and b^j are not in a group. REGROUP combines all the metrics into a single matrix D (Hadamard product) with group candidates. By detecting cycles in the adjacency matrix, we can find connections between people i.e., groups. Then, we employ Depth First Search to detect cycles in an adjacency matrix as commonly done [54]. Thus, potential groups detections $G \in \mathbb{R}^{1xN_t}$ are defined as pedestrians within a cycle and assigned a cluster ID.

B. State Estimation

We consider the group tracking problem for a mobile robot that works in real-world environments to support teams which requires robots to track their teammates from both a stationary and mobile platform. Recent work using Kalman Filters (KF) shows great promise to predict pedestrian states under high egomotion [59]. It models dynamics and uncertainty in learned latent space and performs long-term forecasting [60], [61]. While prior work shows that KFs achieve good tracking performance of pedestrians, there's a lack of work that employs KFs for groups. We adopt the track handling and KF mechanism from [58] for pedestrians and build on it to track groups. We assume that no camera calibration or egomotion information is available and the robot must detect and track groups solely from its onboard sensor (i.e., RGB).

In preliminary experiments, we found that a constant velocity model and linear observation model achieved better group tracking performance than the comparative tracking methods. Thus, we employ these models in our work. States are updated using a linear velocity model when no detection is assigned to a track. The track state is updated using the BB detection.

We represent pedestrian tracks $k \in K$ with an eight-dimensional state space $\langle p_x^k, p_y^k, p_\gamma^k, p_h^k, p_{\dot{x}}^k, p_{\dot{y}}^k, p_{\dot{\gamma}}^k, p_{\dot{h}}^k \rangle$ which is the x, y, aspect ratio, height, and their respective velocities. We compute a set of pedestrian detections $l \in L$ at each timestep and merge the bounding boxes of pedestrians in groups using Equ. 1 to generate group detections $u \in U$.

$$b_u \leftarrow \bigcup_{l \in u} b_l \tag{1}$$

We represent group tracks $v \in V$ with an eight-dimensional state space $\langle g_x^v, g_y^v, g_\gamma^v, g_h^v, g_{\dot{x}}^v, g_{\dot{y}}^v, g_{\dot{\gamma}}^v, g_{\dot{h}}^v \rangle$ which is the x, y, aspect ratio, height, and their respective velocities. Pedestrian and group tracks are initiated when they are observed for three consecutive image frames. We introduce a new technique which stores a pedestrian's group track ID history in $p_{hist} = \{h_t | t = -1, -2, \ldots, -T\}$ to be used for crowd handling where T is the window size (See II-C). When tracks are not observed for A_{max} frames they are removed from track history. We use $A_{max} = 100$ which achieved good performance in empirical experiments. Tracks that are successfully associated for the first three frames continue to be tracked until they exit the frame.

We employ the Hungarian algorithm (HA) to solve the assignment problem for groups, as commonly done in multiple object tracking (MOT) [58], [40], [62]. We define c as a metric which incorporates two distance metrics into the HA to represent motion and appearance (see Eq. 2 from [58]). The motion metric, m, which is standard in KF, tracks the state of a group's position on the image plane over time and generally performs well when egomotion uncertainty is low (i.e., a stationary sensor). The appearance metric a computes the cosine distance between appearance descriptors of group detections u and group tracks v that are generated from a person re-identification CNN (see II-D). We use parameter $\lambda \in$ [0,1] which is the fraction of detections that have been matched to group tracks using the appearance metric a. Next, we employ an indicator function c_{ind} which finds tracks admissible when they are within the gating region of both the appearance metric a_{ind} and motion metric m_{ind} (see Equs. 3 and 4). In practice, REGROUP associates tracks using a and matches the remaining tracks using m.

$$c(u, v) \leftarrow (1 - \lambda) m(u, v) + \lambda a(u, v) \tag{2}$$

$$c_{ind}(u, v) \leftarrow m_{ind}(u, v) \times a_{ind}(u, v) \tag{3}$$

$$m_{ind}, a_{ind} \leftarrow \begin{cases} 1, & \text{if } u \text{ and } v \text{ in the same gating region} \\ 0, & \text{otherwise} \end{cases} \tag{4}$$

We define the bounding box coordinates of group detection u as $b_u = \langle x, y, \gamma, h \rangle$ which are the (x, y) coordinates representing the center of the bounding box, aspect ratio, and height respectively. Also, we define y_v as the track distribution which is a vector of the mean values of the bounding box coordinates for group track v. To compute the motion metric m, we use the squared Mahalanobis distance between new group detections b_u and y_v with inverse covariance S_v^{-1}. The benefit of using Mahalanobis distance is that it computes the $z - score$ statistic, which standardizes the distribution to a mean of 0 and a standard deviation of 1 (See Equ. 5 from [58]). Thus, this property makes it easy to compare the distance from one distribution to another and it captures motion characteristics of groups. m follows the χ^2 distribution with four degrees of freedom (Recall: observation model uses bounding boxes coordinates $\langle x, y, \gamma, h \rangle$) with a critical value of 0.05. Thus, we apply an indicator function m_{ind} to m which assigns detections to tracks by thresholding the Mahalanobis distance at 95% confidence interval computed from inverse χ^2 distribution which results in $\tau^{(1)} = 9.4877$.

$$m(u, v) \leftarrow (b_u - y_v)^T S_v^{-1} (b_u - y_v) \tag{5}$$

$$m_{ind}(u, v) \leftarrow \mathbb{1}[m(u, v) \leq \tau^{(1)}] \tag{6}$$

We define the appearance descriptor of pedestrian detection l as ap_l where $||ap_l|| = 1$ which is the Euclidean norm of the feature vector generated by the CNN defined in Section 3.4. We collect a gallery of appearance descriptors for pedestrians which we defined as $AP_k = \{ap_k^i | i = 1, \ldots, A_k\}$ where A_k is the number of appearance descriptors for the pedestrian track k. Also, we introduce a new appearance descriptor ag_u as the descriptor for group detection u where $||ag_u|| = 1$ which is the Euclidean norm generated by combining the appearance descriptors of pedestrians ap_l in u (See Equ. 7). We collect a gallery of appearance descriptors for groups which we defined as $AG_v = \{ag_v^i | i = 1, \ldots, A_v\}$ where A_v is the number of appearance descriptors for group track v.

$$ag_u \leftarrow \sum_{\substack{l=1 \\ for l \in u}}^{N_t} ap_l \tag{7}$$

To account for camera motion, we use an appearance distance metric a which performs well when egomotion uncertainty is high, such as in mobile robotics applications. First, REGROUP matches pedestrian detection l to pedestrian track k using the cosine distance between their respective appearance descriptors, denoted a^* (See Equ. 8). This metric keeps track of how a pedestrian's appearance changes over time, even after long moments of occlusion, which is useful for recovering tracks when camera motion is dynamic. Then, the system performs a new technique for group data association. It combines the appearance descriptors on pedestrian tracks within group tracks to generate the appearance descriptor of group tracks ag_v (See Equ. 9).

$$a^* \leftarrow \underset{i}{\operatorname{argmin}} \{1 - ap_l^T ap_k^{(i)} | ap_k^{(i)} \in AP_k\} \tag{8}$$

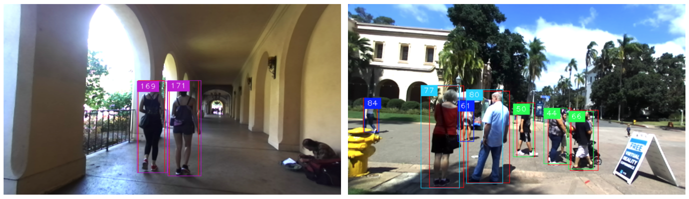

Fig. 4: Example instances of REGROUP ($\eta = 4$) when the scene is not crowded (CIF=0, shown in the left) and it is crowded (CIF=1, shown in the right).

$$ag_v \leftarrow \sum_{\substack{k=1 \\ \forall k \in v}}^{N_t} ap_k^{a^*} \tag{9}$$

Next, REGROUP computes the cosine distance between group detections u and group tracks v using a similar mechanism for the pedestrian appearance metric. Then, we employ an indicator function a_{ind} which assigns group detections to group tracks within the gating region (See Equ. 11. In practice, we make this selection using the group detection that produces the minimum cosine similarity from Equ. 10 [58].

$$a(u,v) \leftarrow min\{1 - ag_u^T ag_v^{(i)} | ag_v^{(i)} \in AG_v\} \tag{10}$$

$$a_{ind}(u,v) \leftarrow \mathbb{1}[a(u,v) \leq \tau^{(2)}] \tag{11}$$

C. *Crowd Indication Feature (CIF)*

When large groups gather together, they are often passing by each other. This typically results in group track mismatches because the ego-centric distance metrics (i.e., D_w, D_h, and D_{gp}) converge for short periods of time. By detecting this situation, we can leverage past track states to preserve group tracks over time. To mitigate this, we introduce a Crowd Indication Feature (CIF) $\in \{0, 1\}$ which detects when the scene is crowded (See Figure 4). We use D_w from Section 3.1 to detect when multiple groups are in close proximity to each other. Then, we apply β to D_w which generates an adjacency matrix. Next, we run depth first search (DFS) on D_w to find cycles, and we select the largest cycle/group, which has a cardinality denoted G_w.

When $G_w < \eta$, this indicates that a crowd is not detected; otherwise, a crowd is detected. We use parameter η to indicate the number of people which constitute a crowd. When conducting experiments with our ego-centric dataset (See Section 5.3.1), we found that $\eta = 4$ generates the best performance. This is likely the result of possible combinations of the number of people passing in groups (e.g., 1-3, 2-2, 1-1-2, etc). When the scene is crowded, we use the past group track history p_{hist} to update group states with a window size T to find the most frequently seen group track IDs. We found that a window size of $T = 10$ timesteps generates good tracking performance.

D. *Deep Appearance Descriptors for Group Data Association*

There has been recent work that aims to re-identify groups of people over time. For instance, [63] uses multi-grain object representations to characterize the appearance and spatial attributes of individuals and subgroups of two people. Multi-grain objects are individual people and subgroups of two and three people inside a group image. [64] uses sparse dictionary learning to transfer knowledge from person re-identification to group re-identification. However, most recent techniques assume that group detections are provided *a priori*, and do not address the problem of data association. Data association is important for group tracking systems because they enable systems to model how group composition changes over time.

REGROUP addresses both of these gaps. First, it learns group detections using our ego-centric group detection system (Discussed in Section 3.1). Then, REGROUP uses a tracking-by-detection technique to track groups of people over time.

A novel strength of REGROUP is that it introduces a new way to do group re-ID to address the group data association problem. The intuition is that a person's appearance does not change much from one timestep to the next. Therefore, by leveraging appearance descriptors, we can design our system to handle high uncertainty due to camera egomotion. Here, we use a person re-ID Convolutional Neural Network (CNN) which is well-suited to re-identify people for data association.

We used a CNN that learns appearance descriptors that discriminate well between people (pretrained on [59]). The CNN architecture consists of two 3x3 convolution layers, a 3x3 max pooling layer, six 3x3 residual layers, a dense layer, followed by a batch and ℓ_2 normalization layer which projects features onto a unit hypersphere [65], [59]. It takes RGB pixels within a pedestrian detection as input and outputs a 1x128 appearance descriptor which we use for data association.

REGROUP generates an appearance descriptor for groups using the appearance descriptor of pedestrians. As mentioned in Section 3.2, we keep a gallery of a pedestrian's appearance descriptors for K timesteps. REGROUP searchers for the best appearance descriptor for each pedestrian in a group (See Equ. 8). Then, it merges the CNNs activation maps of people within groups and uses this as an appearance metric to track groups of people over time (See Equ. 9).

Finally, REGROUP employs a matching cascades (MC) technique [58] to solve the measurement-to-track associations for pedestrians. MC is used to solve the measurement-to-track associations for groups by giving priority to more frequently seen groups when it matches tracks to detections. It starts by computing an association matrix and admissible associations between tracks and detections as defined in Equ. 2 and 3. Then, it solves the linear assignment problem by iterating over the most recent tracks to least recent tracks followed by solving for and updating the matched and unmatched tracks. We conducted all of our experiments on a Dell Inspiron Intel Core i7 laptop, with 16GB RAM, 1TB HDD, and NVIDIA GeForce GTX960M GRU. The machine ran Ubuntu 16.04. We implemented our framework in Python using Tensorflow.

III. Experiments

A. *Dataset Acquisition*

To evaluate REGROUP, we required an ego-centric video dataset captured from a mobile robot in a real-world environ-

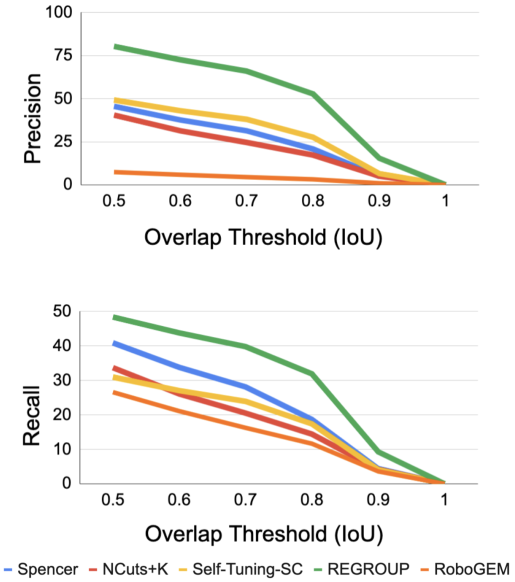

Fig. 5: Group detection performance. Precision (left) and recall (right) measured over IoU threshold (higher is better).

ment (See Figure 6 for an example). There are many publically available group tracking datasets [66], [67], [68]. However, they do not provide spatio-temporal group track annotations, and many are taken from stationary, exocentric cameras.

We augmented the dataset collected by [47] to include additional challenging observations of groups, including different levels of elevation and dynamic pedestrian motion. We mounted a ZED Stereo vision sensor onto a Double Telepresence Robot, which we teleoperated in a public place. The location contained crowds, as well as shaded and open areas[1]. The collection site contains several key computer vision challenges including indoor and outdoor observations, varying lighting conditions, and different degrees of crowdedness.

The total dataset contained 28,094 RGB-D images which we split into training (12,000), validation (6,000), and testing (6,000) sets. The entire dataset contains 8118 unique pedestrians and 52 unique group tracks.

In order to generate group track IDs and bounding boxes, we adopted the definition of groups from [54]. Three members from our team labeled our data using Image Labeler, a built-in MATLAB 2017b application. We validated our labels in a manner consistent with other popular benchmarks in computer vision (e.g. COCO dataset [69], [70]). Thus, two of our team members labeled 3,000 randomly sampled batches of images for label validation. We use Intersection-over-Union (IoU)[2] to evaluate the quality of our labels. Our validation procedure resulted in precision of 78.2 and recall of 71.5 with an IoU of 0.5 which is comparable to COCO's expert annotators.

B. *Experimental Metrics*

We evaluate REGROUP using the widely used standard classification of events, activities, and relationships (CLEAR) multiple object tracking (MOT) metrics [70].

- Multiple Object Tracking Accuracy (MOTA ↑): combines false positives, false negatives, and ID switches.
- Multiple Object Tracking Precision (MOTP ↑): misalignment in BBs between ground truth and predicted tracks.
- Mostly Tracked Targets (MT ↑): number of ground truth BBs covered by a track hypothesis at least 80% of time.
- Mostly Lost Targets (ML ↓): number of ground truth BBs covered by track hypothesis for at most 20% of time.
- False Positives (FP ↓): number of false positives in BBs.
- False Negatives (FN ↓): number of false negatives in BBs.
- Total Number of ID Switches in BBs (IDsw ↓).
- Runtime (t(s) ↓): total time to run a system.

C. *Comparison to State-of-the-Art*

We seek to compare REGROUP to state-of-the-art methods and investigate how well our system performs in terms of group detection and group tracking. To facilitate this, we follow evaluation procedures from [47] which employs similar metrics such as precision and recall in terms of group detection performance. Additionally, we follow the tracking evaluation procedures from [71], [72], [40] which employ metrics to demonstrate our system's ability to track groups long-term. While group detection metrics indicate the performance on a frame-by-frame basis, the group tracking metrics indicate how well the group tracking methods perform long-term.

We tested the group detection and tracking methods independently to evaluate their performance on our challenging dataset. Furthermore, we conducted ablation studies across all methods to evaluate the effectiveness of these methods when different group detection and tracking methods are combined.

1) Group Detection. We compared REGROUP's group detection method against three group detection methods:

Normalized Cuts (NCuts) [73] group pedestrians based on proximity until it reaches K partitions (denoted NCuts+K). We used an off-the-shelf implementation from [73]. We conducted empirical experiments and found that NCuts+K perform best with K=2, so we report those results.

Self-Tuning Spectral Clustering (Self-Tuning-SC) builds on NCuts+K by predicting the number of groups at time t. It predicts K by solving for the eigenvectors and using the eigenvalues to generate K. We created our own implementation[3] following the method presented in [74].

Spencer Group Detector detects groups using social relation features of pedestrians including position, speed, and direction of motion. Then, it trains a Support Vector Machine (SVM) to perform binary classification to detect if two pedestrians are in a group. We created our own implementation following the method presented in [54].

RoboGEM Group Detector [75] clusters pedestrians into groups using agglomerative hierarchical clustering [47], [75]

[1] ZED has 20 meters max range and runs at 20 fps at 640x360 resolution

[2] IOU measures the overlap ratio between two bounding boxes. An IoU of 1 means two boxes perfectly overlap and an IoU of 0 means no overlap.

[3] For many of these methods, they either did not have publicly available code or had implementations we could not get to work. However, we spent months carefully following the methods presented in the papers to ensure a fair comparison.

Group Detector	Group Tracker	Precision ↑	Recall ↑	MT ↑	ML ↓	FP ↓	FN ↓	IDsw ↓	MOTA ↑	MOTP ↑	t(s) ↓
Group-LSTM Tracker											
Spencer	Group-LSTM	30.5	22.8	1	96	4781	7105	284	-32.3	65.9	2254.4
NCuts+K	Group-LSTM	19.6	2.5	0	152	949	8969	18	-7.9	60.7	129.2
Self-Tuning-SC	Group-LSTM	48.4	4.4	0	144	**428**	8800	**0**	-0.3	**67.2**	336.1
REGROUP	Group-LSTM	**55.1**	**29.1**	**2**	**87**	2181	**6521**	282	**2.4**	67.1	**94.5**
Group-LSTM-Obst Tracker											
Group Detector	**Group Tracker**	**Precision ↑**	**Recall ↑**	**MT ↑**	**ML ↓**	**FP ↓**	**FN ↓**	**IDsw ↓**	**MOTA ↑**	**MOTP ↑**	**t(s) ↓**
Spencer	Group-LSTM-Obst	34.8	26.3	2	100	4534	6784	291	-26.2	67.2	2267.0
NCuts+K	Group-LSTM-Obst	21.3	2.6	0	152	896	8958	11	-7.2	62.6	110.3
Self-Tuning-SC	Group-LSTM-Obst	55.8	5.1	0	146	**370**	8734	**0**	1.1	67.5	356.5
REGROUP	Group-LSTM-Obst	**66.4**	**35.1**	**4**	**79**	1634	**5974**	307	**14.0**	**71.2**	**94.7**
Spencer Tracker											
Group Detector	**Group Tracker**	**Precision ↑**	**Recall ↑**	**MT ↑**	**ML ↓**	**FP ↓**	**FN ↓**	**IDsw ↓**	**MOTA ↑**	**MOTP ↑**	**t(s) ↓**
Spencer	Spencer	45.3	40.1	5	62	4454	5509	**264**	-11.2	75.7	2309.8
NCuts+K	Spencer	39.0	33.1	4	68	4766	6149	459	-23.7	74.2	131.5
Self-Tuning-SC	Spencer	45.9	29.2	8	67	**3163**	6507	399	-9.5	77.7	340.4
REGROUP	Spencer	**54.0**	**60.2**	**35**	**34**	4717	**3656**	466	**3.9**	**78.7**	**117.1**
REGROUP Tracker											
Group Detector	**Group Tracker**	**Precision ↑**	**Recall ↑**	**MT ↑**	**ML ↓**	**FP ↓**	**FN ↓**	**IDsw ↓**	**MOTA ↑**	**MOTP ↑**	**t(s) ↓**
Spencer	REGROUP	63.8	26.6	1	91	1388	6752	**112**	10.3	77.0	2377.4
NCuts+K	REGROUP	66.4	29.4	6	75	1369	6487	190	12.5	77.0	269.0
Self-Tuning-SC	REGROUP	71.1	24.5	8	88	**918**	6938	144	13.0	77.7	494.3
REGROUP	REGROUP	**80.4**	**48.4**	**27**	**54**	1090	**4744**	197	**34.5**	**80.7**	**132.4**

TABLE I: Ablation Study Findings with group detectors and group trackers (See Section III-C).

with features such as pedestrian position, velocity, orientation, and distance from the robot to people in the environment.

2) **Group Tracking.** We compared REGROUP's group tracking method against three group tracking methods:

Group-LSTM segments pedestrians by clustering trajectories of individuals that have similar motion trends. It tracks groups using an Long-Short Term Memory Recurrent Neural Network network to predict the motion of the pedestrians. We used an off-the-shelf implementation by [55].

Group-LSTM-Obst builds on Group-LSTM by predicting the future motion trajectory of pedestrians after n timesteps ($n >= 1$) by leveraging grouping behaviors and obstacles in the environment. Group-LSTM-Obst leverages the layout of the environment to predict the trajectory of groups over time. We used an off-the-shelf implementation by [56].

Spencer's Group Tracker uses a Multi-Hypothesis Tracker to track groups over time from an ego-centric perspective. We used an off-the-shelf implementation by [54] [4].

We conducted ablation experiments to understand how different group detection methods impact the performance of the group tracking methods. Here, we explored the combination of the group tracking methods, including Group-LSTM [55], Group-LSTM-Obst [56], and REGROUP with the group detection methods including NCuts+K [73], Spencer [54], Self-Tuning-SC [74], and REGROUP's group detector.

IV. Results

A. Group Detection

Figure 5 shows the overall group detection results. Overall, REGROUP outperformed all other methods by up to 40% in terms of precision and up to 18% in terms of recall. Self-Tuning-SC outperformed Spencer, and Spencer outperformed NCuts+K in terms of precision. Spencer outperformed NCuts+K and Self-Tuning-SC in terms of recall. For $IoU < 0.5$, NCuts+K outperformed Self-Tuning-SC, but for IoU ≥ 0.6, Self-Tuning-SC performed better than NCuts+K. REGROUP's detector outperforms RoboGEM's detector in terms of precision by up to 30% and recall by up to 18%.

B. Group Tracking

Table I shows the group tracking results. Overall, REGROUP outperformed all other methods by up to 66% in terms of MOTA and 20% in terms of MOTP.

Group-LSTM Tracking Table I shows the Group-LSTM ablation results. Overall performance of Group-LSTM improves using REGROUP's group detection method in terms of precision, recall, MT, ML, and FN, MOTA, and MOTP. Also, the performance declines using the Self-Tuning-SC group detection method in terms of across all metrics except FP. The performance declines further using the Spencer group detection method in terms of precision, FP, IDsw, and MOTA; although it achieves better performance than NCuts+K and Self-Tuning-SC in terms of recall, MT, ML, FN, and MOTP. Lastly, the performance of Group-LSTM using NCuts+K achieves the poorest performance of all group detection methods. In terms of runtime, the Group-LSTM tracker achieves the shortest total runtime using REGROUP's group detector, and the longest runtime using the Spencer group detector.

Group-LSTM-Obst Tracking Table I shows the Group-LSTM-Obst results. Overall performance of Group-LSTM-Obst improves using REGROUP's group detection method in terms of precision, recall, MT, ML, FN, MOTA, and MOTP. The performance of Group-LSTM-Obst declines using Self-Tuning-SC in terms of all metrics except FP. Also, the performance declines using Spencer group detection method in terms of precision, FP, IDsw, and MOTA; although, the performance improves in terms of recall, MT, FN, and MOTP compared to NCuts+K and Self-Tuning-SC methods. The performance of Group-LSTM-Obst declines most using NCuts+K. In terms of

[4] https://github.com/spencer-project/spencer_people_tracking

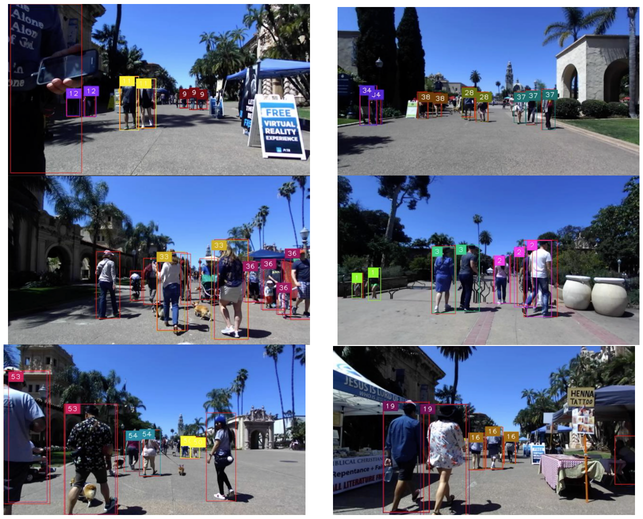

Fig. 6: Example REGROUP results on our dataset.

runtime, Group-LSTM-Obst achieves the shortest total runtime using REGROUP's group detector.

Spencer Tracking Table I shows the Spencer ablation results. Overall performance of Spencer improves using REGROUP's group detection method in terms of precision, recall, MT, ML, FN, MOTA, and MOTP. The performance of Spencer declines using Self-Tuning-SC and Spencer group detectors in terms of all metrics except FP and IDsw. Also, the performance declines using NCuts+K group detection method in terms of precision, recall, MT, ML, FP, IDsw, MOTA, and MOTP; although, the performance improves in terms of recall, FN, and MOTP compared to the Self-Tuning-SC method. In terms of runtime, Spencer achieves the shortest total runtime using REGROUP's group detector.

REGROUP Tracking Table I shows the REGROUP's ablation results. Overall REGROUP achieved the best tracking performance when compared to Group-LSTM methods [55], [56] across all metrics except FP and IDsw. The performance of Spencer declines in terms of precision, MT, ML, and MOTA performance which is likely due to its poor group detection performance (See Figure 5). NCuts+K outperforms Spencer across all metrics. Lastly, our method achieves the shortest total runtime using REGROUP's group detector.

V. Discussion

In this work, we introduced REGROUP, a group detection and tracking system for mobile robots working in real-world environments. We demonstrated that deep learning appearance descriptors have the potential to address the group detection and tracking problem. Even with no *a priori* knowledge (i.e., for training group detectors), REGROUP outperforms Spencer [54], Self-Tuning-SC [74], NCuts+K [73], Group-LSTM [55], and Group-LSTM-Obst [56] methods in terms of group detection and tracking on our dataset.

Our ablation studies showed how well both the group detector and tracker of REGROUP performed in comparison to the other methods. For instance, Figure 5 shows that REGROUP achieves the best group detection performance in terms of precision and recall across all methods. REGROUP's detector also improves the performance of all other tracking methods in terms of precision, recall, MT, ML, FN, MOTA, and MOTA, including Spencer, Group-LSTM, Group-LSTM-Obst and REGROUP (See Table I). Also, Self-Tuning-SC achieves the second best performance in terms of precision, which also reflects the results of this method when combined with Group-LSTM and Group-LSTM-Obst group tracking methods. Spencer achieves the second best performance in terms of recall, MT, ML, FN, and MOTP.

As shown in Table I, Group-LSTM and Group-LSTM-Obst performance declines with MT=0 using NCuts+K and Self-Tuning-SC group detection methods. This indicates that Group-LSTM and Group-LSTM-Obst methods do not track any groups for at least 80% of the time they are observed in the scene. This is likely caused by inconsistent group detections where groups are detected in one frame and not the next frame which results in generating a new group track ID even when groups are detected in the future. As a consequence, Group-LSTM methods often swapped IDs between groups.

All tracking methods achieve the shortest runtime using REGROUP's group detector. As aforementioned, REGROUP achieves the best performance of all group tracking methods. This shows that REGROUP is beneficial because it achieves a shorter runtime without sacrificing precision, recall, or MOTA.

Our group tracking approach is beneficial to HRI in several ways. First, REGROUP is able to track groups in real-world, human-centered environments where people are moving and the robot is moving, a well-known problem in robotics. Second, it uses group data association that leverages deep learning features, which enable robots to leverage appearance features when egomotion uncertainty is high [76], [77], [52]. Third, REGROUP can enable navigation systems operating in human-centered environments to engage in more socially aware interaction with human groups.

In the future, we plan to continue building on REGROUP to reach our goal of designing robots that safely and fluently with groups. There are many exciting domains to deploy REGROUP, such as in retail settings, work sites, and in hospitals, to support teams. We are particularly interested in deploying REGROUP in conjunction with navigation systems to enable robots to support human teams in safety critical environments. For instance, robots can use REGROUP to track healthcare workers as it works alongside them (e.g., delivering supplies, helping patients stand), and can be helpful in other teaming contexts such as manufacturing and search and rescue [78], [79]. Furthermore, this work is useful in other areas of robotics, such as in last-mile and personal transportation applications [6], where understanding what groups of people are doing can enable robots to make intelligent decisions.

To help support reproducability, code for REGROUP can be found at: https://github.com/UCSD-RHC-Lab/regroup-public.

We hope this work will prove useful for the HRI community, as it contributes a new system to investigate how groups move throughout an environment and it can enable robots to seamlessly work in human-robot teaming situations.

References

[1] R. Triebel, K. Arras, R. Alami, L. Beyer, S. Breuers, R. Chatila, M. Chetouani, D. Cremers, V. Evers, M. Fiore *et al.*, "Spencer: A socially aware service robot for passenger guidance and help in busy airports," in *Field and Service Robotics*. Springer, 2016, pp. 607–622.

[2] Y. Lee, S. Lee, and D.-Y. Kim, "Exploring hotel guests' perceptions of using robot assistants," *Tourism Management Perspectives*, vol. 37, p. 100781, 2021.

[3] L. D. Riek, "Healthcare robotics," *Communications of the ACM*, pp. 68–78, 2017.

[4] A. Taylor, H. R. Lee, A. Kubota, and L. D. Riek, "Coordinating clinical teams: Using robots to empower nurses to stop the line," *Proceedings of the ACM on Human-Computer Interaction*, vol. 3, no. CSCW, pp. 1–30, 2019.

[5] H. Bai, S. Cai, N. Ye, D. Hsu, and W. S. Lee, "Intention-aware online pomdp planning for autonomous driving in a crowd," in *IEEE Intern. Conf. on Robotics and Automation (ICRA)*. IEEE, 2015, pp. 454–460.

[6] G. Nichols, "The last mile: Robots take to streets for local delivery," 2021. [Online]. Available: https://www.zdnet.com/article/the-last-mile-robots-take-to-streets-for-local-delivery/

[7] D. M. Chan and L. D. Riek, "Unseen salient object discovery for monocular robot vision," *IEEE Robotics and Automation Letters*, vol. 5, no. 2, pp. 1484–1491, 2020.

[8] D. Chan, A. Taylor, and L. D. Riek, "Faster robot perception using salient depth partitioning," in *Proceedings of the Intern. Conf. on Intelligent Robots and Systems (IROS)*. IEEE, 2017, pp. 4152–4158.

[9] J. Oh, T. M. Howard, M. R. Walter, D. Barber, M. Zhu, S. Park, A. Suppe, L. Navarro-Serment, F. Duvallet, A. Boularias *et al.*, "Integrated intelligence for human-robot teams," in *Intern. Symposium on Experimental Robotics*. Springer, 2016, pp. 309–322.

[10] M. E. Napoli, H. Biggie, and T. M. Howard, "On the performance of selective adaptation in state lattices for mobile robot motion planning in cluttered environments," in *2017 IEEE/RSJ Intern. Conf. on Intelligent Robots and Systems (IROS)*. IEEE, 2017, pp. 4436–4443.

[11] U. Patel, N. Kumar, A. J. Sathyamoorthy, and D. Manocha, "Dynamically feasible deep reinforcement learning policy for robot navigation in dense mobile crowds," *arXiv preprint arXiv:2010.14838*, 2020.

[12] A. J. Sathyamoorthy, J. Liang, U. Patel, T. Guan, R. Chandra, and D. Manocha, "Densecavoid: Real-time navigation in dense crowds using anticipatory behaviors," in *2020 IEEE Intern. Conf. on Robotics and Automation (ICRA)*. IEEE, 2020, pp. 11 345–11 352.

[13] A. Fishman, C. Paxton, W. Yang, D. Fox, B. Boots, and N. Ratliff, "Collaborative interaction models for optimized human-robot teamwork," in *2020 IEEE/RSJ Intern. Conf. on Intelligent Robots and Systems (IROS)*. IEEE, 2020, pp. 11 221–11 228.

[14] A. Rasouli and J. K. Tsotsos, "Autonomous vehicles that interact with pedestrians: A survey of theory and practice," *IEEE transactions on intelligent transportation systems*, vol. 21, no. 3, pp. 900–918, 2019.

[15] J. S. Brar and B. Caulfield, "Impact of autonomous vehicles on pedestrians' safety," in *2017 IEEE 20th Intern. Conf. on Intelligent Transportation Systems (ITSC)*. IEEE, 2017, pp. 714–719.

[16] A. Vemula, K. Muelling, and J. Oh, "Social attention: Modeling attention in human crowds," in *2018 IEEE Intern. Conf. on Robotics and Automation (ICRA)*. IEEE, 2018, pp. 4601–4607.

[17] T. Iqbal and L. D. Riek, "Human robot teaming: Approaches from joint action and dynamical systems," in *Humanoid robotics: A reference*, 2019, pp. 2293–2312.

[18] M. Vázquez, A. Steinfeld, and S. Hudson, "Parallel detection of conversational groups of free-standing people and tracking of their lower-body orientation," in *2015 IEEE/RSJ Intern. Conf. on Intelligent Robots and Systems (IROS)*. IEEE, 2015, pp. 3010–3017.

[19] M. R. Fraune, "Our robots, our team: Robot anthropomorphism moderates group effects in human–robot teams," *Frontiers in Psychology*, vol. 11, 2020.

[20] K. Liaw, S. Driver, and M. R. Fraune, "Robot sociality in human-robot team interactions," in *Intern. Conf. on Human-Computer Interaction*. Springer, 2019, pp. 434–440.

[21] M. R. Fraune, S. Šabanović, and T. Kanda, "Human group presence, group characteristics, and group norms affect human-robot interaction in naturalistic settings," *Frontiers in Robotics and AI*, vol. 6, p. 48, 2019.

[22] E. André, A. Paiva, J. Shah, and S. Šabanovic, "Social agents for teamwork and group interactions (dagstuhl seminar 19411)," in *Dagstuhl Reports*, vol. 9, no. 10. Schloss Dagstuhl-Leibniz-Zentrum fuer Informatik, 2020.

[23] F. Yang, W. Yin, M. Björkman, and C. Peters, "Impact of trajectory generation methods on viewer perception of robot approaching group behaviors," in *2020 29th IEEE Intern. Conf. on Robot and Human Interactive Communication (RO-MAN)*. IEEE, 2020, pp. 509–516.

[24] F. Yang, Y. Gao, R. Ma, S. Zojaji, G. Castellano, and C. Peters, "A dataset of human and robot approach behaviors into small free-standing conversational groups," *PloS one*, vol. 16, no. 2, p. e0247364, 2021.

[25] T. Kruse, A. K. Pandey, R. Alami, and A. Kirsch, "Human-aware robot navigation: A survey," *Robotics and Autonomous Systems*, vol. 61, no. 12, pp. 1726–1743, 2013.

[26] A. Taylor and L. D. Riek, "Robot perception of human groups in the real world: State of the art," in *AAAI Fall Symposium Series: Artificial Intelligence for Human-Robot Interaction Technical Report*, vol. 4, 2016, p. 2017.

[27] F. Yücel, Z.and Zanlungo, T. Ikeda, T. Miyashita, and N. Hagita, "Deciphering the crowd: Modeling and identification of pedestrian group motion," *Sensors*, vol. 13, no. 1, pp. 875–897, 2013.

[28] L. Bazzani, M. Cristani, and V. Murino, "Decentralized particle filter for joint individual-group tracking," in *IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*. IEEE, 2012, pp. 1886–1893.

[29] A. Al Masum, M. H. Rafy, and S. M. Rahman, "Video-based affinity group detection using trajectories of multiple subjects," in *Intern. Conf. on Electrical and Computer Engineering (ICECE)*. IEEE, 2014, pp. 120–123.

[30] H. Yu, Y. Zhou, J. Simmons, C. P. Przybyla, Y. Lin, X. Fan, Y. Mi, and S. Wang, "Groupwise tracking of crowded similar-appearance targets from low-continuity image sequences," in *Proceedings of the IEEE Conf. on Computer Vision and Pattern Recognition*, 2016, pp. 952–960.

[31] L. Bazzani, M. Zanotto, M. Cristani, and V. Murino, "Joint individual-group modeling for tracking," *IEEE Transactions on Pattern Analysis & Machine Intelligence*, no. 4, pp. 746–759, 2015.

[32] C. Garate, S. Zaidenberg, J. Badie, and F. Bremond, "Group tracking and behavior recognition in long video surveillance sequences," in *Intern. Conf. on Computer Vision Theory and Applications (VISAPP)*, vol. 2. IEEE, 2014, pp. 396–402.

[33] S. D. Khan, G. Vizzari, S. Bandini, and S. Basalamah, "Detection of social groups in pedestrian crowds using computer vision," in *Intern. Conf. on Advanced Concepts for Intelligent Vision Systems*. Springer, 2015, pp. 249–260.

[34] Y. Zhang, L. Qin, S. Zhang, H. Yao, and Q. Huang, "Formation period matters: Towards socially consistent group detection via dense subgraph seeking," in *Proceedings of the 5th ACM on Intern. Conf. on Multimedia Retrieval*. ACM, 2015, pp. 475–478.

[35] N. Li, Y. Zhang, W. Luo, and N. Guo, "Instant coherent group motion filtering by group motion representations," *Neurocomputing*, vol. 266, pp. 304–314, 2017.

[36] L. Leal-Taixé, G. Pons-Moll, and B. Rosenhahn, "Everybody needs somebody: Modeling social and grouping behavior on a linear programming multiple people tracker," in *Intern. Conf. on Computer Vision Workshops (ICCV)*. IEEE, 2011, pp. 120–127.

[37] P. Stefano, E. Andreas, S. Konrad, and L. Van Gool, "You'll never walk alone: Modeling social behavior for multi-target tracking," in *Intern. Conf. on Computer Vision Workshops (ICCV)*, 2009.

[38] R. Mazzon, F. Poiesi, and A. Cavallaro, "Detection and tracking of groups in crowd," in *Intern. Conf. on Advanced Video and Signal Based Surveillance (AVSS)*. IEEE, 2013, pp. 202–207.

[39] O. H. Jafari, D. Mitzel, and B. Leibe, "Real-time rgb-d based people detection and tracking for mobile robots and head-worn cameras," in *Intern. Conf. on Robotics and Automation (ICRA)*. IEEE, 2014, pp. 5636–5643.

[40] T. Linder, S. Breuers, B. Leibe, and K. O. Arras, "On multi-modal people tracking from mobile platforms in very crowded and dynamic environments," in *Intern. Conf. on Robotics and Automation (ICRA)*. IEEE, 2016, pp. 5512–5519.

[41] J. S. Smith, R. Xu, and P. Vela, "egoteb: Egocentric, perception space navigation using timed-elastic-bands," in *2020 IEEE Intern. Conf. on Robotics and Automation (ICRA)*. IEEE, 2020, pp. 2703–2709.

[42] W. Choi, Y. Chao, C. Pantofaru, and S. Savarese, "Discovering groups of people in images," in *European Conf. on computer vision*. Springer, 2014, pp. 417–433.

[43] D. Brščić, F. Zanlungo, and T. Kanda, "Modelling of pedestrian groups and application to group recognition," in *40th Intern. Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO)*. IEEE, 2017, pp. 564–569.

[44] M. Luber and K. O. Arras, "Multi-hypothesis social grouping and tracking for mobile robots." in *Robotics: Science and Systems (RSS)*, 2013.

[45] R. J. Sethi, "Towards defining groups and crowds in video using the atomic group actions dataset," in *Intern. Conf. on Image Processing (ICIP)*. IEEE, 2015, pp. 2925–2929.

[46] I. Chatterjee and A. Steinfeld, "Low cost perception of dense moving crowd clusters for appropriate navigation," in *Workshop on Social Norms in Robotics and HRI, IEEE/RSJ Intern. Conf. on Intelligent Robots and Systems (IROS)*, 2015.

[47] A. Taylor, D. Chan, and L. Riek, "Robot-centric perception of human groups," *Transactions on Human Robot Interaction*, vol. 9, no. 3, pp. 1–21, 2020.

[48] M. Hashimoto, A. Tsuji, A. Nishio, and K. Takahashi, "Laser-based tracking of groups of people with sudden changes in motion," in *Intern. Conf. on Industrial Technology (ICIT)*. IEEE, 2015, pp. 315–320.

[49] P. Bergmann, T. Meinhardt, and L. Leal-Taixe, "Tracking without bells and whistles," in *Proceedings of the IEEE/CVF Intern. Conf. on Computer Vision*, 2019, pp. 941–951.

[50] J. Xu, Y. Cao, Z. Zhang, and H. Hu, "Spatial-temporal relation networks for multi-object tracking," in *Proceedings of the IEEE/CVF Intern. Conf. on Computer Vision*, 2019, pp. 3988–3998.

[51] J. Zhu, H. Yang, N. Liu, M. Kim, W. Zhang, and M. Yang, "Online multi-object tracking with dual matching attention networks," in *Proceedings of the European Conf. on Computer Vision (ECCV)*, 2018, pp. 366–382.

[52] A. Sadeghian, A. Alahi, and S. Savarese, "Tracking the untrackable: Learning to track multiple cues with long-term dependencies," pp. 300–311, 2017.

[53] S. Tang, M. Andriluka, B. Andres, and B. Schiele, "Multiple people tracking by lifted multicut and person re-identification," in *Proceedings of the IEEE Conf. on Computer Vision and Pattern Recognition*, 2017, pp. 3539–3548.

[54] T. Linder and K. O. Arras, "Multi-model hypothesis tracking of groups of people in rgb-d data," in *7th Intern. Conf. on Information Fusion (FUSION)*. IEEE, 2014, pp. 1–7.

[55] N. Bisagno, B. Zhang, and N. Conci, "Group lstm: Group trajectory prediction in crowded scenarios," in *Proceedings of the European Conf. on computer vision (ECCV)*, 2018, pp. 0–0.

[56] N. Bisagno, C. Saltori, B. Zhang, F. G. De Natale, and N. Conci, "Embedding group and obstacle information in lstm networks for human trajectory prediction in crowded scenes," *Computer Vision and Image Understanding*, p. 103126, 2020.

[57] J. Redmon and A. Farhadi, "Yolov3: An incremental improvement," *arXiv preprint arXiv:1804.02767*, 2018.

[58] N. Wojke, A. Bewley, and D. Paulus, "Simple online and realtime tracking with a deep association metric," in *IEEE Intern. Conf. on Image Processing (ICIP)*. IEEE, 2017, pp. 3645–3649.

[59] N. Wojke and A. Bewley, "Deep cosine metric learning for person re-identification," in *2018 IEEE winter Conf. on applications of computer vision (WACV)*. IEEE, 2018, pp. 748–756.

[60] V. L. Guen and N. Thome, "Disentangling physical dynamics from unknown factors for unsupervised video prediction," in *Proceedings of the IEEE/CVF Conf. on Computer Vision and Pattern Recognition*, 2020, pp. 11 474–11 484.

[61] L. Zhou, Z. Luo, T. Shen *et al.*, "Kfnet: Learning temporal camera relocalization using kalman filtering," in *Proceedings of the IEEE/CVF Conf. on Computer Vision and Pattern Recognition*, 2020, pp. 4919–4928.

[62] W. Luo, J. Xing, A. Milan, X. Zhang, W. Liu, X. Zhao, and T. Kim, "Multiple object tracking: A literature review," *Artificial Intelligence*, p. 103448, 2020.

[63] H. Xiao, W. Lin, B. Sheng, K. Lu, J. Yan, J. Wang, E. Ding, Y. Zhang, and H. Xiong, "Group re-identification: Leveraging and integrating multi-grain information," in *Proceedings of the 26th ACM Intern. Conf. on Multimedia*, 2018, pp. 192–200.

[64] G. Lisanti, N. Martinel, A. Del Bimbo, and G. Luca Foresti, "Group re-identification via unsupervised transfer of sparse features encoding," in *Proceedings of the IEEE Intern. Conf. on Computer Vision*, 2017, pp. 2449–2458.

[65] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE Conf. on computer vision and pattern recognition*, 2016, pp. 770–778.

[66] R. B. Fisher, "The pets04 surveillance ground-truth data sets," in *Proceedingings of 6th IEEE Intern. workshop on performance evaluation of tracking and surveillance*, 2004, pp. 1–5.

[67] M. Moussaïd, N. Perozo, S. Garnier, D. Helbing, and G. Theraulaz, "The walking behaviour of pedestrian social groups and its impact on crowd dynamics," *PloS one*, vol. 5, no. 4, p. e10047, 2010.

[68] F. Bartoli, G. Lisanti, L. Seidenari, S. Karaman, and A. Del Bimbo, "Museumvisitors: a dataset for pedestrian and group detection, gaze estimation and behavior understanding," in *Proceedings of the IEEE Conf. on Computer Vision and Pattern Recognition Workshops*, 2015, pp. 19–27.

[69] T. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick, "Microsoft coco: Common objects in context," in *European Conf. on Computer Vision*. Springer, 2014, pp. 740–755.

[70] 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.

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

[72] T. Linder, F. Girrbach, and K. O. Arras, "Towards a robust people tracking framework for service robots in crowded, dynamic environments," in *Assistance and Service Robotics Workshop (ASROB-15) at the IEEE/RSJ Intern. Conf. on Intelligent Robots and Systems (IROS)*, 2015.

[73] J. Shi and J. Malik, "Normalized cuts and image segmentation," *IEEE Transactions on pattern analysis and machine intelligence*, vol. 22, no. 8, pp. 888–905, 2000.

[74] L. Zelnik-manor and P. Perona, "Self-tuning spectral clustering," in *Advances in Neural Information Processing Systems*, L. Saul, Y. Weiss, and L. Bottou, Eds., vol. 17. MIT Press, 2005. [Online]. Available: https://proceedings.neurips.cc/paper/2004/file/40173ea48d9567f1f393b20c855bb40b-Paper.pdf

[75] A. Taylor and L. D. Riek, "Robot-centric human group detection," in *13th Annual ACM/IEEE Intern. Conf. on Human-Robot Interaction, Social Robots in the Wild Workshop*. IEEE, 2018.

[76] L. Leal-Taixé, A. Milan, K. Schindler, D. Cremers, I. Reid, and S. Roth, "Tracking the trackers: an analysis of the state of the art in multiple object tracking," *arXiv preprint arXiv:1704.02781*, 2017.

[77] K. Fang, Y. Xiang, and S. Savarese, "Recurrent autoregressive networks for online multi-object tracking," *IEEE Winter Conf. on Applications of Computer Vision (WACV)*, 2017.

[78] S. Matsumoto, S. Moharana, N. Devanagondi, L. Oyama, and L. Riek, "Iris: A low-cost telemedicine robot to support healthcare safety and equity during a pandemic," in *Proceedings of the 15th Conf. on Pervasive Computing Technologies for Healthcare (PervasiveHealth)*, 2021.

[79] A. Taylor, S. Matsumoto, W. Xiao, and L. Riek, "Social navigation for mobile robots in the emergency department," in *IEEE/RSJ Intern. Conf. on Intelligent Robots and Systems (IROS)*. IEEE, 2021.