Robust Pedestrian Tracking in Crowd Scenarios using an Adaptive GMM-based Framework

Shuyang Zhang[1], Di Wang[2], Fulong Ma[3], Chao Qin[1], Zhengyong Chen[1] and Ming Liu[3]

***Abstract*— In this paper, we address the issue of pedestrian tracking in crowd scenarios. People in close social relationships tend to act as a group which is a great challenge to individually discriminate and track pedestrians on a LiDAR system. In this paper, we integrally model groups of people and track them in a recursive framework based on Gaussian Mixture Model (GMM). The model is optimized by an extended Expectation-Maximization (EM) algorithm which can adaptively vary the number of mixture components over scans. Experimental results both qualitatively and quantitatively indicate the reliability and accuracy of our tracker in populated scenarios.**

I. INTRODUCTION

A. Motivation

Multi-object tracking (MOT) is crucial for mobile robotics applications. The host robot monitors the complicated motion of its surroundings and provides reliable motion estimates for subsequent decision-making module. It is especially intractable in crowd scenarios, such as factory and campus, where pedestrians are the major participants and have special interactive behaviors [1] [2] [3]. People in intimate social relationships tend to be in close physical position and normally crowd or walk as a group in public.

The first challenge of tracking groups of people is to extract individuals. As shown in Fig. 1, friends walk as a group and have small physical intervals. It is a hardship for spatial segmentation methods to cluster each individual from a LiDAR scan. The second challenge is to keep the association consistency under the interaction of other group members. The distance between two close pedestrians keep changing over scans. Sometimes, they can be separately segmented but in some cases are wrongly considered as one. This uncertainty in segmentation makes data association ambiguous and ultimately causes the degradation of tracking performance.

Tracking groups of people are extensively studied in camera-based [4] [5] [6], LiDAR-based [2] [3] [7] and fusion-based [8] [9] system. An inspiring thought is to model individuals and people groups separately [2] [7] [10] and they proposed that modeling a group of people as a single object is more efficient. Pedestrians and people groups are respectively clustered and modeled in reasonable representation, such as contours [2] and Gaussian shapes [10]. Afterward, they are tracked within Bayesian filtering frameworks. Lau *et al.* [2] also creatively discussed the interconversion and interaction between individuals and groups.

Fig. 1. Hybrid view of a crowd scenario. The point cloud is collected by four RS-LiDAR-16 scanners on our intelligent delivery robot. People marked by the yellow box are in close proximity and it is challenging to cluster each individual using spatial segmentation methods.

Similar to the work of Mucientes *et al.* [10], we model pedestrians by using Gaussian shapes. The difference is that we also model the entire LiDAR scan as a GMM combined with a uniform clutter model. The number of Gaussian components, which represents the number of pedestrians in the scene, is supported by the prior model and it can be adaptively updated using our extended EM algorithm.

B. Contribution

In this paper, we addressed this issue of individual pedestrian tracking in crowd scenarios. The contributions of this paper are summarized as follows:

- A GMM-based framework focused on tracking individuals in crowd scenarios.
- An extended EM algorithm for GMM parameter learning which adaptively changes the number of Gaussian components.
- Quantitative and qualitative evaluation with real scene data which shows robust tracking performance under people's interaction.

[1] Shuyang Zhang, Chao Qin and Zhengyong Chen are with Unity-Drive technology Inc, Shenzhen, China {shuyang.zhang, chao.qin, chenzhengyong}@unity-drive.com.

[2] Di Wang is with Institute of Artificial Intelligence and Robotics, Xian Jiaotong University, China {de2wang}@stu.xjtu.edu.cn.

[3] Fulong Ma and Ming Liu are with the Robotics and Multi-Perception Laborotary, Robotics Institute, The Hong Kong University of Science and Technology, Hong Kong SAR, China. {fmaaf, eelium}@ust.hk.

C. Organization

The remainder of this paper is structured as follows. Sect. II systematically introduces state-of-the-art work on LiDAR-based tracking. Sect. IV-A presents the mixture model while Sect. IV-B details our extended EM algorithm to solve this aforementioned model. Sect. IV-C introduces details in our tracking pipeline. Our tracker is evaluated in Sect. V while Sect. VI concludes our method and future work.

II. Related Works

LIDAR-based multi-object tracking has been researched for decades. We categorize them as model-free and model-based methods.

A. Model-free Methods

Model-free methods can generally adapt to objects with various shapes and sizes. Firstly, They extract objects from a LiDAR scan in the form of clusters using spatial cues, such as distance [11] [12] and angle [13], or using motion cues [14] [15]. Later, observation is aligned with existing tracks via their position or feature similarity. Finally, the track state of each object is independently or jointly updated under a Bayesian filtering framework. These methods using spatial cues are implemented under an intuitive pipeline. However, the performance of the data association is highly related to the prior segmentation procedures. Under- and over-segmentation frequently occur and alignment ambiguity is brought into data association, which ultimately degrades tracking performance.

Methods using motion cues [14] distinguish objects via their movement difference and is rarely affected by association ambiguity. However, pedestrians in group usually share similar movement and motion-based methods can not perfectly separate individuals.

B. Model-based Methods

In model-based methods, objects are detected and associated with the knowledge of priori geometric models. For vehicle tracking, Petrovskaya *et al.* [16] used a rectangle measurement model in their likelihood-field-based framework while Fortin *et al.* [17] proposed a polyhedron-like one. For pedestrians tracking, Shackleton *et al.* [3] proposed a pedestrian surface model. They collect a set of contour descriptions on people's surface over scans and align them using a surface matching technique. Spinello *et al.* [18] proposed a sliced pedestrian model. They subdivide a person into multiple layers defined by height and learn a specialized classifier for each layer. Mucientes *et al.* [10] modeled groups of people as Gaussian shapes. They consider data association, clusters split and join within an MHT framework.

Our work is a model-based one and we make an assumption that people can be represented as Gaussian shapes. The difference is that we also integrally model an entire LiDAR scan by a mixture model. The number of mixture components is adaptively updated over scans.

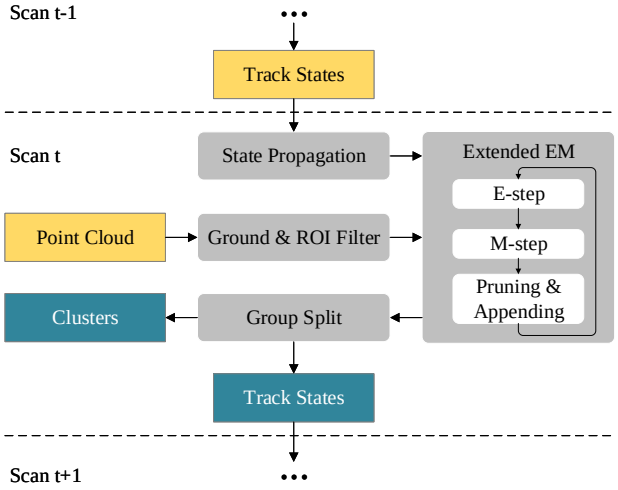


Fig. 2. System pipeline of the proposed method. The system inputs of each scan are the previous track states and the incoming point cloud. After the state propagation and point cloud filtering, the inputs are used to initialize a GMM-based mixture model which is then optimized by our extended EM algorithm. Group split is afterward considered and the final outputs are the cluster results and the updated track states.

III. Overview

The pipeline of our pedestrian tracking framework is shown in Fig. 2. We assume that our robot benefits from a good localization system, so the uncertainty of ego-motion can be negligible over the time scale of the whole track.

RS-LiDAR-16 scanners, which are used in our LiDAR system and suffer from a poor vertical resolution of 1°, are not sufficient for 3D segmentation. Meanwhile, our intelligent delivery robot is deployed on the road surface and objects in the workspace can be constrained on the 2D plane. Like other autonomous driving applications [19] [20], points are converted to 2D representation in the top-down coordinate system.

IV. Methodology

A. GMM-based Scan Modeling

After ground and ROI filtering, a LiDAR scan can be represented as

$$\boldsymbol{\mathcal{X}} = [\boldsymbol{x}_1, ..., \boldsymbol{x}_i, ..., \boldsymbol{x}_N] \in \mathbb{R}^{2\times N}, \tag{1}$$

where N denotes the number of points which are in 2D format. Inspired by Horaud *et al.* [21], we model $\boldsymbol{\mathcal{X}}$ as a Gaussian mixture model with an uniform distribution

$$p(\boldsymbol{x}_i) = \sum_{k=1}^{K} \pi_k \mathcal{N}(\boldsymbol{x}_i | \boldsymbol{\mu}_k, \boldsymbol{\sigma}_k) + \frac{\pi_0}{h}. \tag{2}$$

The number of Gaussian components K represents the amount of observed objects in the scan. $\boldsymbol{\mu_k}$ and $\boldsymbol{\sigma_k}$ are the parameters of the k-th Gaussian component and π_k denotes the mixture coefficient where $\sum_{k=0}^{K} \pi_k = 1$. The parameters of our model can be denoted as

$$\boldsymbol{\Theta} = \{\{\pi_k\}_{k=0}^{K}, \{\boldsymbol{\mu_k}\}_{k=1}^{K}, \{\boldsymbol{\sigma_k}\}_{k=1}^{K}\}. \tag{3}$$

The clutter is modeled as a uniform distribution and its coefficient π_0 is manually set to control the proportion of outliers. h represents the area of 2D workspace which contains all the data points. π_0 and h mutually determine the stability of Gaussian shapes and we will give a detailed discussion in Sect. V-B.

Traditional methods choose K empirically. It is difficult to directly determine the number of pedestrians in a scan. We initialize the first scan by DBSCAN [22], a spatial segmentation method which does not need a selected K. GMM parameters in the current scan propagate and are used to initialize the mixture model in the next scan.

B. Extended Expectation-Maximization algorithm

The parameter learning of the Gaussian mixture model is solved via an extended EM algorithm which can be subdivided into 3 parts: Expectation, Maximization, Pruning and Appending. For each iteration, the first two steps update hidden variables $\boldsymbol{Z}$ and model parameters $\boldsymbol{\Theta}$ while the last one considers the variation of component number K.

The hidden variables are defined as $\boldsymbol{Z} = \{z_i\}_{i=1}^N$, where $z_i \in \{0, 1, 2, ..., K\}$ indicates the component that $\boldsymbol{x}_i$ belongs to. $\boldsymbol{Z}$ can also be considered as the segmentation results of the scan $\boldsymbol{\mathcal{X}}$. The posterior $\gamma_{ik} = p(z_i = k|\boldsymbol{x}_i)$ denotes the probability that point $\boldsymbol{x}_i$ aligns to the k-th component. The set of γ_{ik} is an $N \times (K + 1)$ matrix and is denoted as $\boldsymbol{\Gamma}$.

1) Expectation: The posteriors $\boldsymbol{\Gamma}$, are updated by mixture model parameters $\boldsymbol{\Theta}$. For each posterior

$$\gamma_{ik} = \begin{cases} \frac{1}{\eta}\pi_k \mathcal{N}(\boldsymbol{x_i}|\boldsymbol{\mu_k}, \boldsymbol{\sigma_k}), & k \neq 0, \\ 1 - \sum_{j=1}^{K} \frac{1}{\eta}\pi_j \mathcal{N}(\boldsymbol{x_i}|\boldsymbol{\mu_j}, \boldsymbol{\sigma_j}), & k = 0, \end{cases} \tag{4}$$

where $\eta = \sum_{j=1}^{K} \pi_j \mathcal{N}(x_i|\boldsymbol{\mu_j}, \boldsymbol{\sigma_j}) + \frac{\pi_0}{h}$.

2) Maximization: Mixture model parameters $\boldsymbol{\Theta}$ are separately updated by posteriors $\boldsymbol{\Gamma}$. The log-likelihood function of Eq.2 is

$$\mathcal{L}(\boldsymbol{\mathcal{X}}|\boldsymbol{\Theta}) = \sum_{i=1}^{N} \ln(\sum_{k=1}^{K} \pi_k \mathcal{N}(x_i|\boldsymbol{\mu_k}, \boldsymbol{\sigma_k}) + \frac{\pi_0}{h}). \tag{5}$$

GMM parameters are respectively calculated by using standard optimization techniques [23] and we do not restate the details. One point of which needs to be reminded is the update of mixture coefficients. As a uniform distribution is embedded into the mixture model, the sum of Gaussian coefficients needs to be subjected to $1 - \pi_0$.

3) Pruning and Appending: In this step, we consider the variation of component number K. We prune the Gaussian components of exiting objects and append the incoming ones.

For pruning, we use the Gaussian degeneration in GMM to judge whether a component needs to be removed. We also use the other two criteria via the mixture model coefficients and the relevant point number. We define an indicator function $\mathbb{I}(\cdot)$ that is equal to 1 if the input is true and 0 otherwise. The indicator of criteria for the k-th component $\mathbb{I}_k$ is defined as

$$\mathbb{I}_k(\boldsymbol{\mathcal{X}}, \pi_k, \boldsymbol{\mu_k}, \boldsymbol{\sigma_k}) = \prod_{l=1}^{3} \mathbb{I}_{k,l}, \tag{6}$$

where

$$\mathbb{I}_{k,1}(\boldsymbol{\sigma_k}) \triangleq \begin{cases} 1, & |\boldsymbol{\sigma_k}| < d_{min\boldsymbol{\sigma}}, \\ 0, & others, \end{cases} \tag{7}$$

$$\mathbb{I}_{k,2}(\pi_k) \triangleq \begin{cases} 1, & \pi_k < \pi_{min}, \\ 0, & others, \end{cases} \tag{8}$$

$$\mathbb{I}_{k,3}(\boldsymbol{\mathcal{X}}, \boldsymbol{\mu_k}, \boldsymbol{\sigma_k}) \triangleq \begin{cases} 1, & n_r < n_{min}, \\ 0, & others. \end{cases} \tag{9}$$

The threshold n_r in Eq. 9 is the number of relevant points

$$n_r = \sum_{i=1}^{N} \mathbb{I}(\mathcal{N}(\boldsymbol{x_i}|\boldsymbol{\mu_k}, \boldsymbol{\sigma_k}) < \alpha_{0.005}). \tag{10}$$

The k-th component is trimmed when $\mathbb{I}_k = 1$ and $K_{prune} = \sum_{k=1}^{K} \mathbb{I}_k$ is the number of deleted components.

For appending, the points marked as outliers $\mathcal{X}_{z_i=0}$ are segmented by DBSCAN. The K_{append} new clusters are initialized as new Gaussian components. The mixture coefficient of a new component $\pi_{new,j}$ is initialized as $1/(K - K_{prune})$.

Parameters Γ and $\{\pi_k\}_{k=0}^{K}$ need to be renormalized after both pruning and appending because the number of components changes. After each EM iteration, the number of Gaussian components is updated as

$$K^* = K - K_{prune} + K_{append}. \tag{11}$$

C. Tracking Cycle

This part describes the steps in a cycle of our pedestrian tracker and the details are shown in Fig. 2. We use a tracking-before-detection technique. Firstly, our tracker propagates people's position to the current scan. Afterward, it models the scan as a GMM and performs segmentation by using our extended EM algorithm. Finally, split and merge of group people are considered.

1) Track State: We do not implement a general probabilistic filter for pedestrian tracking. The principal reason is the motion of people can not be simply considered as physics-based models. Interactive behaviors in a populated environment make their movement unpredictable.

Our tracker uses the Gaussian shape model in Sect .IV-A directly. We define a track $\boldsymbol{\mathcal{T}}$ as a sequence of measurements over scans which are assumed to derive from the same object. For n-th object, its track state is $\boldsymbol{\mathcal{S}}_n = \{\boldsymbol{\mu}_n, \boldsymbol{\sigma}_n, \boldsymbol{v}_n\}$ and $\boldsymbol{\mathcal{T}}_n = \{\boldsymbol{\mathcal{S}}_{n,t}\}_{t=1}^{T}$. $\boldsymbol{\mu}_n = (x, y)^T$ is the position vector and $\boldsymbol{\sigma}_n \in \mathbb{R}^{2\times2}$ represents the shape of target pedestrian. We need to clarify that in tracking pipeline, the subscripts lose their particular meaning in Sect. IV-A and are only used to indicate specified track. The part of velocity $\boldsymbol{v}_n = (v_x, v_y)^T$ is differentially calculated and smoothed using Kalman smoother.

2) Prediction: At the beginning of a tracking cycle, we propagate previous track states to the current scan and make them the initial values for our GMM-based segmentation. The $\boldsymbol{\mu}_n$ and $\boldsymbol{\sigma}_n$ in track state is predicted as

$$\begin{cases} \bar{\boldsymbol{\mu}}_n = \boldsymbol{\mu}_n + \boldsymbol{v}_n \delta t, \\ \bar{\boldsymbol{\sigma}}_n = \boldsymbol{\sigma}_n + \boldsymbol{R} \delta t, \end{cases} \tag{12}$$

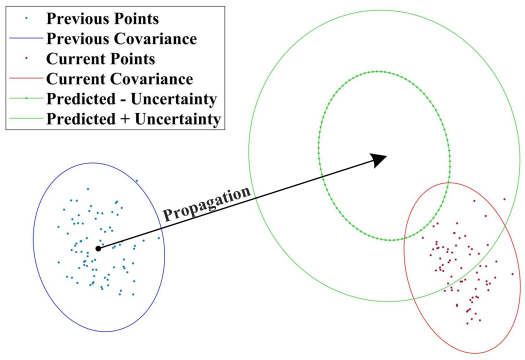


Fig. 3. State Propagation for an object. After adding uncertainty to the prediction step, points are more likely to converge to close-by Gaussian components instead of the clutter. The optimization of GMM may converge incorrectly when directly using the covariances from the former scan. The ellipse denotes the one-sigma Gaussian boundary.

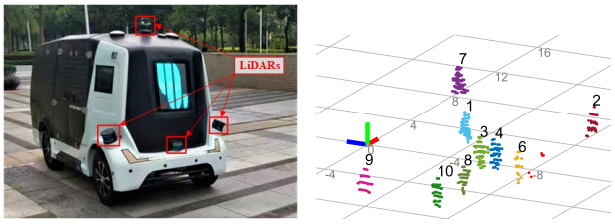
where δt denotes the interval between adjacent scan. $R \in \mathbb{R}^{2\times 2}$ represents a pedestrian's maximum motion in a unit time.

Motion uncertainty can give bad initial values for EM optimization which may lead to mistakes that points converge to incorrect components. We add this motion uncertainty directly into a pedestrian's shape model to improve the stability of EM initialization. In Expectation step, γ_{ik} becomes larger accompanied with the increase of $\boldsymbol{\sigma}_k$ and surrounding points are more likely related to the Gaussian component than outliers. R takes effect on $\boldsymbol{\sigma}_k$ only once for a scan. After the first Maximization step, the covariance will converge to the proper size.

3) Joint Segmentation and Data Association: We use a tracking-before-detection technique to perform segmentation and data association simultaneously. Segmentation results are derived by the optimized mixture model while data association is realized by the propagation and update of model parameters over scans. The details are described in Sect. IV-B.

4) Tracking Management: The entrance of the new objects and exit of the existing ones are also considered by our extended EM algorithm in Sect. IV-B.

We also implement some traditional tracking techniques in our tracker. When our tracker loses an object, we do not discard its track immediately but continue a prediction for 5 scans. Gating is also realized in our tracker. It is adjusted by the clutter model and the details will be specifically discussed in Sect. V-B.



(a) LiDARs configuration on our platform.

(b) A glance of *hard* dataset with ground truth annotation.

Fig. 4. Our intelligent delivery robot and pedestrian tracking dataset.

D. Group Merge and Split

In this part, we discuss the merge of individuals and the split of people groups.

1) Merge: When two pedestrians reunion, they can be separated no matter how close they are because they are already modeled as two Gaussian components in previous scans. These two closed-by persons may not be perfectly divided in point level, but they have already been able to ensure tracking consistency.

2) Split: When two persons are initially modeled as one and walk away from each other in subsequent scans, they are still modeled as one Gaussian distribution, even though they can be spatially separated.

We implement an additional DBSCAN algorithm for the whole scan and its result is only used to validate our GMM-based segmentation. The segmentation output from DBSCAN is represented as $\boldsymbol{C} = \{c_i\}_{i=1}^{N}$, which has a similar representation as $\boldsymbol{Z}$. We define the point index of the $j-$th component in our model as $\boldsymbol{I}_{\boldsymbol{Z}_j} = \{i|z_i = j\}$. For each cluster m in $\boldsymbol{C}$, we calculate their ratio

$$r = \frac{\left|\boldsymbol{I}_{\boldsymbol{C}_m} \cap \boldsymbol{I}_{\boldsymbol{Z}_j}\right|}{\left|\boldsymbol{I}_{\boldsymbol{Z}_j}\right|}, \tag{13}$$

where $|\cdot|$ indicates the cardinality of the index set. If $r_{min} < r < 1$ and $\left|\boldsymbol{I}_{\boldsymbol{C}_m} \cap \boldsymbol{I}_{\boldsymbol{Z}_j}\right| > n_{min}$, we model each $\boldsymbol{I}_{\boldsymbol{C}_m} \cap \boldsymbol{I}_{\boldsymbol{Z}_j}$ as a new Gaussian component.

E. Initialization

An incoming scan is pre-processed by a ground and ROI filter. Ground points are filtered by a Gaussian-Process-Regression-based method [24]. The 3D ROI filter removes points out of our robot's motion space. For the initialization of our tracker's first scan, DBSCAN is implemented to perform a spatial segmentation. DBSCAN is also used in our extended EM algorithm and group people split module.

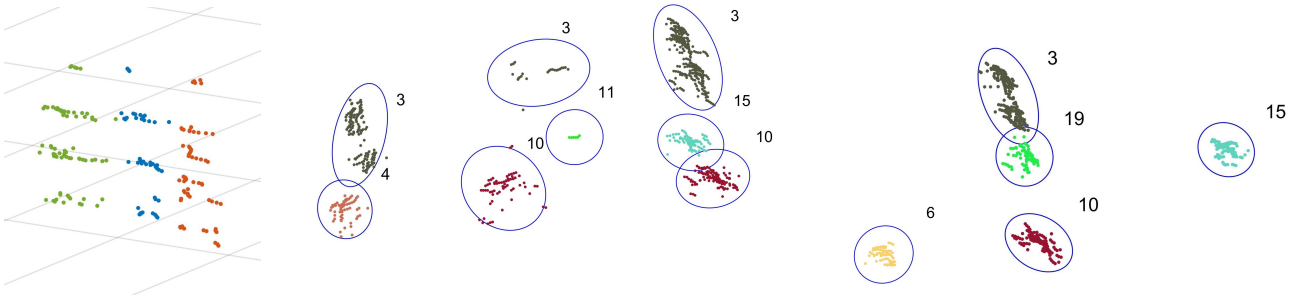
V. EXPERIMENTS

In this section, we introduce the datasets and our selected baseline methods. We also detail the implementation parameters and evaluate our method in tracking performance, cardinality estimates and timing.

A. Datasets

For experimental evaluation, we collected two datasets with different challenge levels. The data is captured by our intelligent delivery robot (Fig. 4(a)) in a populated area of an industrial park. Our LiDAR system is composed of four spinning RS-LiDAR-16 scanners. All LiDARs are hardware synchronized and calibrated [25]. The details of these two datasets are as follows:

- *easy*: A specific dataset that contains 6 independent sequences. There are only several tracks per sequence, but all of these tracks last more than 100 scans. Group



(a) Ground truth (b) Scan 1 (c) Scan 51 (d) Scan 110 (e) Scan 124

Fig. 5. Qualitative analysis of our tracker. Three pedestrians with the same motion are first tracked with ID No.3 and No.4. (b) Our tracker is initialed by DBSCAN which clusters two individuals as one object. (c) Occlusion makes a wrong split and creates a new observation for clutter (No.11). It vanishes in 5 scans but causes an ID switch from No.4 to No.10. (d) A person (No.15) passes across the people of three. Our tracker handles this interactive condition effectively. (e) Another one (No.19) walks by the group. They are too close that their points mix together but our tracker can still separate them. It is important to point out that our tracker can not subdivide the two people on track No.3. Their distance never exceeds the split threshold. However, it is reasonable to track them as one because they share the same motion.

(a) $pi_0 = 0.01$ (b) $pi_0 = 0.001$

Fig. 6. The Gaussian one-sigma ellipse with a different choice of π_0. The man is waving his hand and the shape of point cloud has a great change over scans. When using a larger π_0, points away from the Gaussian center are snatched by the clutter model.

merge and object entrance happen but in a limited number.

- *hard*: A general sequence on a populated street, which lasts for 395 scans and contains 36 tracks and 2667 observations. The major participants are pedestrians but several cyclists are also included. The dataset is more complicated where group merge and split, occlusion, object entrance and exit are fully contained.

We use our initialization procedure in Sect. IV-E to filter ground points and the ones out of ROI. The remained points are hand-labeled for each individuals.

B. Implementation Details

The values of π_0 and h are crucial for modeling. As shown in Fig. 6, the component π_0/h in our model represents the strength of clutter and it affects the actual range of each Gaussian model in segmentation. It can be considered as a gating function which contributes to the association between model components and raw points. We assume that the effective range of RS-LiDAR-16 for detecting pedestrians is $\pm 10m$, so the area of our workspace in the clutter model $h = 400m^2$. Combined with the selection of h, we set $\pi_0 = 0.001$.

The rest of the parameters are determined empirically. EM optimization terminates with a maximum iteration $n_{max} = 10$ and a convergence criterion that the average variation of μ_k is under $0.01m$. In pruning and appending step, $d_{min\boldsymbol{\sigma}} = 0.01$, $\pi_{min} = 0.005$ and $n_{min} = 3$. In DBSCAN algorithm, the minimum number of points is the same as n_{min} and the radius threshold in Sect. IV-E and Sect. IV-B is $0.2m$. The radius threshold of DBSCAN is set as $0.5m$ in Sect. IV-D to balance under-segmentation with occlusions cases. In tracking cycle, pedestrians motion uncertainty $\boldsymbol{R} = diag([1.0, 1.0])$. In group split, the ratio threshold for split people group $r_{min} = 0.3$.

C. Baseline Methods

We implement a segmentation method of Wang *et al.* [11] which is based on Euclidean Minimum Spanning Tree (EMST). As the background points in our dataset have been filtered, we do not achieve foreground extraction in their original work. The threshold for breaking the EMST is set as 0.2m. We make an additional data association module achieved by the Munkres algorithm and each cluster is associated via the similarity of their gravity center.

Spectral Clustering (SC) method uses higher-dimensional features and we suppose it will have a better performance than spatial segmentation on close pedestrians. Spectral Clustering is sensitive to the number of clusters (K in GMM), which needs to be set prudently. We directly initial the cluster number from ground truth to achieve the best performance. We also use the Munkres algorithm for data association.

For further discussion, we do not implement specific pedestrian tracking methods such as [2] [7] [10]. They discuss group people as an independent tracking element and do not try to separate them. These methods will be unfairly evaluated in our point-level annotation.

D. Evaluation: tracking

We evaluate the tracking performance of our method both qualitatively and quantitatively.

The qualitative results are demonstrated in Fig. 5. We analyze a typical group tracking example in our *hard* dataset.

The quantitative results are shown in Table I. We use several metrics [26] [27] which are widely used in multi-object tracking and can evaluate the precision and accuracy of a tracker. The selected MOT metrics are as follows:

TABLE I

EVALUATION:TRACKING

Dataset	Method	MOTA↑	MOTP↓	FN↓	FP↓	IDS↓
easy	EMST	0.800	0.038	213	33	3
	SC	0.936	0.083	25	29	37
	ours	**0.999**	**0.004**	**0**	**2**	**0**
hard	EMST	0.493	0.041	604	655	83
	SC	0.634	0.064	322	392	262
	ours	**0.920**	**0.017**	**156**	**24**	**16**

- MOTP: The average errors between estimated positions and their corresponding ground truth.
- MOTA: The ratio of correct data association and it indicates the ability to keep accurate trajectories.
- FP (false positives): The number of false alarms.
- FN (false negatives): The number of missed detections.
- IDS (identity switches): The number of mismatches.

Our method has the best performance in both datasets, especially in *hard* dataset which contains a certain amount of pedestrians in the scene and complex people interaction. We get a higher MOTA score of 0.920 in *hard* dataset, compared to the SC with 0.634 and the EMST with 0.493. In MOTP score, our method has an average error of 0.017m which is observably lower than the SC with 0.064m and the EMST with 0.041m. It should be noted that compared to FP and IDS, our method suffers more from FN. The *hard* dataset contains special instances that people crowd as a group in the whole sequence (Fig. 5). Our method sometimes clusters and tracks them as one object so the number of missed detections increases.

E. Evaluation: cardinality estimates

We also evaluate the cardinality estimates, which in this paper means the number of predicted pedestrians in each scan. The cardinality estimates of our method and the EMST baseline method are shown in Fig. 7.

Our average error in pedestrian number per scan is 0.4557 while the EMST has 0.4633. The difference is not significant between our method and the EMST baseline, but it is obviously in Fig. 7 that our tracking number changes more gently than the EMST method. This conclusion is also confirmed by the MOTA score in Table I.

F. Evaluation: timing

After ground and ROI filtering, the number of remained points is approximately 2000 which is stress-free for DBSCAN. We set the maximum iterations as 10, but our model optimization is totally converged for an average times of 3.6.

Our method is tested and evaluated on MATLAB R2019a. We also implement a ROS CPP version on a PC with an Intel i5-8600K CPU and 8 GB RAM. The consuming time per scan is averagely 51.47ms, which is sufficient for online use on our platform.

VI. CONCLUSIONS AND FUTURE WORK

In this paper, we proposed a pedestrian tracking framework in crowd scenarios based on Gaussian Mixture Model.

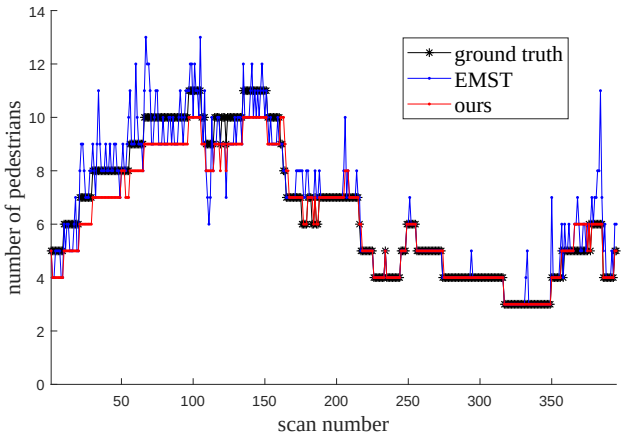


Fig. 7. Relationship between the cardinality estimates and the scan number.

The model is optimized by an extended Expectation-Maximization algorithm which can adaptively vary the number of mixture components over scans. Both qualitative and quantitative evaluation on tracking performance indicates that the proposed method shows robustness and accuracy in populated scenarios.

Our method shows good stability in pedestrian modeling. However, vehicles are sometimes contained in the scene which can be ambiguous in Gaussian representation. One extension of our work is to embed a 3D LiDAR detection CNN into our tracking module. Vehicles can be pre-treated and our framework can be expanded to a more general scenario.

REFERENCES

[1] P. Zhang, W. Ouyang, P. Zhang, J. Xue, and N. Zheng, "Sr-lstm: State refinement for lstm towards pedestrian trajectory prediction," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 12085–12094, 2019.

[2] B. Lau, K. O. Arras, and W. Burgard, "Tracking groups of people with a multi-model hypothesis tracker," in *2009 IEEE International Conference on Robotics and Automation*, pp. 3180–3185, IEEE, 2009.

[3] J. Shackleton, B. VanVoorst, and J. Hesch, "Tracking people with a 360-degree lidar," in *2010 7th IEEE International Conference on Advanced Video and Signal Based Surveillance*, pp. 420–426, IEEE, 2010.

[4] S. J. McKenna, S. Jabri, Z. Duric, A. Rosenfeld, and H. Wechsler, "Tracking groups of people," *Computer vision and image understanding*, vol. 80, no. 1, pp. 42–56, 2000.

[5] A. Setia and A. Mittal, "Co-operative pedestrians group tracking in crowded scenes using an mst approach," in *2015 IEEE Winter Conference on Applications of Computer Vision*, pp. 102–108, IEEE, 2015.

[6] 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 Conference on Computer Vision and Pattern Recognition Workshops*, pp. 19–27, 2015.

[7] G. Gate and F. Nashashibi, "Fast algorithm for pedestrian and group of pedestrians detection using a laser scanner," in *2009 IEEE Intelligent Vehicles Symposium*, pp. 1322–1327, IEEE, 2009.

[8] D. Held, D. Guillory, B. Rebsamen, S. Thrun, and S. Savarese, "A probabilistic framework for real-time 3d segmentation using spatial, temporal, and semantic cues.," in *Robotics: Science and Systems*, 2016.

[9] J. Ku, A. D. Pon, S. Walsh, and S. L. Waslander, "Improving 3d object detection for pedestrians with virtual multi-view synthesis orientation estimation," *arXiv preprint arXiv:1907.06777*, 2019.

[10] M. Mucientes and W. Burgard, "Multiple hypothesis tracking of clusters of people," in *2006 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 692–697, IEEE, 2006.

[11] D. Z. Wang, I. Posner, and P. Newman, "What could move? finding cars, pedestrians and bicyclists in 3d laser data," in *2012 IEEE International Conference on Robotics and Automation*, pp. 4038–4044, IEEE, 2012.

[12] K. Klasing, D. Wollherr, and M. Buss, "A clustering method for efficient segmentation of 3d laser data," in *2008 IEEE International Conference on Robotics and Automation*, pp. 4043–4048, IEEE, 2008.

[13] I. Bogoslavskyi and C. Stachniss, "Fast range image-based segmentation of sparse 3d laser scans for online operation," in *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 163–169, IEEE, 2016.

[14] A. Dewan, T. Caselitz, G. D. Tipaldi, and W. Burgard, "Motion-based detection and tracking in 3d lidar scans," in *2016 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 4508–4513, IEEE, 2016.

[15] D. Z. Wang, I. Posner, and P. Newman, "Model-free detection and tracking of dynamic objects with 2d lidar," *The International Journal of Robotics Research*, vol. 34, no. 7, pp. 1039–1063, 2015.

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

[17] B. Fortin, R. Lherbier, and J.-C. Noyer, "A model-based joint detection and tracking approach for multi-vehicle tracking with lidar sensor," *IEEE Transactions on Intelligent Transportation Systems*, vol. 16, no. 4, pp. 1883–1895, 2015.

[18] L. Spinello, K. O. Arras, R. Triebel, and R. Siegwart, "A layered approach to people detection in 3d range data," in *Twenty-fourth AAAI conference on artificial intelligence*, 2010.

[19] M. He, E. Takeuchi, Y. Ninomiya, and S. Kato, "Precise and efficient model-based vehicle tracking method using rao-blackwellized and scaling series particle filters," in *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 117–124, IEEE, 2016.

[20] H. Cho, Y.-W. Seo, B. V. Kumar, and R. R. Rajkumar, "A multi-sensor fusion system for moving object detection and tracking in urban driving environments," in *2014 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 1836–1843, IEEE, 2014.

[21] R. Horaud, F. Forbes, M. Yguel, G. Dewaele, and J. Zhang, "Rigid and articulated point registration with expectation conditional maximization," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 33, no. 3, pp. 587–602, 2011.

[22] M. Ester, H.-P. Kriegel, J. Sander, X. Xu, *et al.*, "A density-based algorithm for discovering clusters in large spatial databases with noise.," in *Kdd*, vol. 96, pp. 226–231, 1996.

[23] C. M. Bishop, *Pattern recognition and machine learning*. springer, 2006.

[24] T. Chen, B. Dai, R. Wang, and D. Liu, "Gaussian-process-based real-time ground segmentation for autonomous land vehicles," *Journal of Intelligent and Robotic Systems*, vol. 76, no. 3, pp. 563–582, 2014.

[25] J. Jiao, Y. Yu, Q. Liao, H. Ye, R. Fan, and M. Liu, "Automatic calibration of multiple 3d lidars in urban environments," in *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 15–20, Nov 2019.

[26] K. Bernardin, A. Elbs, and R. Stiefelhagen, "Multiple object tracking performance metrics and evaluation in a smart room environment," in *Sixth IEEE International Workshop on Visual Surveillance, in conjunction with ECCV*, vol. 90, p. 91, Citeseer, 2006.

[27] H. Caesar, V. Bankiti, A. H. Lang, S. Vora, V. E. Liong, Q. Xu, A. Krishnan, Y. Pan, G. Baldan, and O. Beijbom, "nuscenes: A multimodal dataset for autonomous driving," *arXiv preprint arXiv:1903.11027*, 2019.