



Knowledge Acquisition from Pedestrian Flow Analysis using Sparse Mobile Probe Data

Ranulfo Plutarco Bezerra Neto¹ · Kazunori Ohno^{2,3} · Thomas Westfechtel⁴ · Shotaro Kojima¹ · Kento Yamada¹ · Satoshi Tadokoro¹

Received: 30 May 2020 / Accepted: 7 May 2021 / Published online: 17 July 2021
© The Author(s) 2021

Abstract

Autonomous vehicles require high-level semantic maps, which contain the activities of pedestrians and cars, to ensure safe navigation. High-level semantics can be obtained from mobile probe sensor data. Analyzing pedestrian trajectories obtained from mobile probe data is an effective approach to avoid collisions between autonomous vehicles and pedestrians. Such analyses of pedestrian trajectories can generate new information such as pedestrian behaviors in violation of traffic regulations. However, pedestrian trajectories obtained from mobile probe data significantly sparse and noisy, making it challenging to analyze pedestrian activity. To address this issue, we propose multiple daily data and graph-based approaches to treat sparse and noisy data for estimating the flow of pedestrians based on mobile probe data. To improve the sparseness of the data, multiple daily data are fused. After that, a pedestrian graph is created to enhance the region's coverage by connecting the sparse data indicating the flow of pedestrians. This proposed approach successfully obtained pedestrian trajectory data from the sparse and noisy data. Moreover, it was possible to identify the potential locations where pedestrians tend to cross the street by analyzing the pedestrian flow. The results indicate that 83% of well-known regions where pedestrians tend to cross the street corresponded with those extracted using the proposed approach. Furthermore, a high-level semantic map of the regions where pedestrians tend to cross the street along a 1-km road is presented. The trajectory information obtained using the proposed approach is expected to be essential for understanding different scenarios of the interactions between individuals and autonomous vehicles.

Keywords Knowledge acquisition · Mobile probe data · Trajectory analysis

This work was partially supported by JST CREST Recognition, Summarization, and Retrieval of Large-Scale Multimedia Data Grant Number JP14532298, NEDO's commissioned research project, Integration at the core of next-generation artificial intelligence and robots / 1. Research and development, Demonstration for social implementation of artificial intelligence technology / Research and development of transportation of soil at local small and medium construction sites using robot technology and artificial intelligence", and AIP Challenge Research Funding, Japan.

✉ Ranulfo Plutarco Bezerra Neto
bezerra.ranulfo@rm.is.tohoku.ac.jp

- ¹ Tohoku University, Graduate School of Information Science, Sendai, Japan
- ² New Industry Creation Hatchery Center, Sendai, Japan
- ³ RIKEN Center for Advanced Intelligence Project, Sendai, Japan
- ⁴ The University of Tokyo, Research Center for Advanced Science and Technology, Tokyo, Japan

1 Introduction

Currently, several companies are attempting to develop autonomous vehicles. However, several challenges need to be resolved to develop a fully autonomous vehicle. One major challenge faced by such autonomous systems is the unpredictability of pedestrian movement in traffic zones. Pedestrians often tend to ignore traffic signs and cross the streets in the absence of crosswalks [1]. To help resolve this problem, this study aims to extract high-level information from the movement of pedestrians.

Autonomous vehicles require considerable information and data to achieve sufficient situational awareness for navigating through unpredictable urban environments involving humans traveling in cars, bikes, and other vehicles as well as pedestrians. To navigate in such environments, autonomous cars employ semantic maps to understand their surrounding environments. These semantic maps contain information

regarding the environment's structure, such as roads, traffic signs, and sidewalks. Moreover, these maps should also include dynamic information of the scene, such as vehicle and pedestrian movement. Thus, this dynamic information can be used to understand the rules of the scene, such as locations commonly traversed by pedestrians. Identifying the areas where pedestrians tend to cross the street despite the absence of a crosswalk, enables autonomous vehicles to drive more carefully (or slowly) in such areas; this yields additional reaction time to the vehicle if a pedestrian decides to cross the street. This approach can help prevent accidents, particularly if the pedestrian was previously occluded from the autonomous vehicle's sensors, by parked vehicles or other visual obstructions. In this study, we employ an autonomous vehicle to extract the dynamic information of a scene. This information is further used to create a map that represents the pedestrian crossing patterns within a scene.

However, to extract pedestrian crossing information, several problems need to be addressed. First, due to the short period for which pedestrians are visible by a probe vehicle driving within the street's speed limit, the pedestrian information extracted from the scene is insufficient. Second, due to this insufficient pedestrian information extracted from the scene, the pedestrian data obtained by the vehicle are sparse. Third, the extracted pedestrian scan points can also contain miss-labeled information due to sensor noise.

This paper presents an approach for trajectory-based information acquisition from one mobile probe vehicle's multiple daily data. We present a method to detect pedestrians in an environment, track the pedestrians, and generate a general flowing graph based on their movements. Based on this graph, we extract regions wherein people are more likely to cross the street. In the proposed approach, an autonomous electric vehicle equipped with a Ladybug camera and Velodyne HDL-32E LiDAR sensors is employed to gather data. As the vehicle is driven at the recommended speed during data acquisition, we were able to cover a large area; however, the data obtained were sparse. To resolve this sparseness in the pedestrian data, we employed three different methods: First, we fused multiple daily data. Second, we utilized clustering and a graph-based approach to connect pedestrian regions. Third, a set of heuristics were used to improve the graph connection. Then, the improved graph is analyzed, and the regions wherein pedestrians tend to cross the street are extracted. Finally, based on the extracted crossing regions, a map of pedestrian crossing regions is created. Figure 1 shows an illustration of the proposed study.

This study is an extended version of [2], which introduces the creation of pedestrian flow. We improved this previous implementation of the pedestrian flow using multiple daily data and a novel clustering approach. Moreover, this paper outlines a method to extract information from this

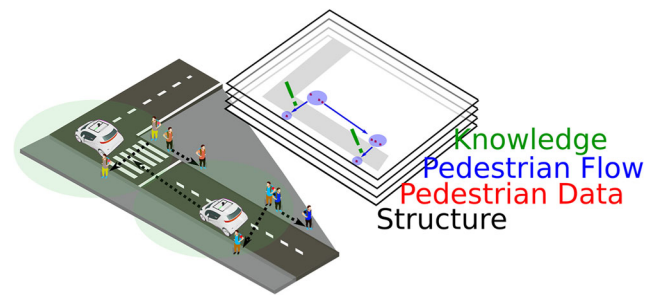


Fig. 1 Illustration of knowledge acquisition using pedestrian flow

pedestrian flow using multiple daily data obtained using one mobile probe. This extracted information is further used to develop a map indicating the regions where pedestrians are likely to cross the street.

The primary contributions of this study are as follows:

- Here, the trajectories are estimated from sparse mobile probe data. The use of tracking, a graph-based approach, and multiple daily data enables the recovery of pedestrian trajectories.
- Pedestrian walking direction is obtained from the pedestrian scan points, using a clustering approach to extract trajectory information of the cluster.
- The proposed approach predicts pedestrian crossing patterns based on semantics as well as pedestrian and vehicle flow.

2 Related Works

2.1 Mobile Probe Data

Several studies have employed mobile probe data to extract trajectories, such as GPS data, signal-based data, stationary sensor-based data, and mobile sensor-based data. GPS data yields [3, 4] location information, sparse and irregularly distributed in time. Moreover, using this data is challenging because using GPS data can result in serious privacy breaches.

Signal-based data can be classified into short-range distance [5, 6] and long-range distance [7, 8]. Short-range distance extraction requires base stations such as WiFi routers or IoT equipment to receive Bluetooth or radio-frequency identification signals. The range of data acquisition depends on the number of base stations. Long-range distance employs GSM or CDR base stations to extract signals from a broader range. However, data extracted through these signal-based methods tend to be complex and differ depending on the type of signal used; moreover, these methods' location accuracy is relatively low.

Stationary sensor-based data refers to the data acquired through stationary monitor systems such as cameras or Automatic Fare Collection (AFC) [9–11]. Although data

extracted through such sensors exhibit relatively high accuracy, data retrieval's amplitude is small due to the limited range of the stationary sensors.

The works of [12–15] employ mobile sensor-based data using a vehicle equipped with cameras or light detection and ranging (LiDAR) sensors to extract the trajectories of traffic participants. Mobile sensor-based data features high position accuracy as well as broad coverage area. Therefore, this study employs mobile sensor-based data captured through LiDAR and a camera sensor of an autonomous electric vehicle (AEV). The combination of LiDAR and camera used in this study extracts the precise 3D scan of the environment, with semantic and color information.

2.2 Pedestrian Flow

Pedestrian flow is commonly used to understand pedestrian behavior and can also extract information from pedestrian movement. For instance: [16] used pedestrian trajectories to reconstruct a network of streets, [17] used pedestrian flow to estimate the congestion and danger in an area, [18] used pedestrian and vehicle flow to model traffic light scheduling strategies for vehicles as well as pedestrians, [19] used pedestrian flow to create an autonomous navigation system for a personal mobility vehicle, and [20] used pedestrian flow to improve long term prediction of the pedestrian path. Such studies have reported a variety of applications demonstrating the wide applicability of pedestrian flow information. However, none of the previous studies have employed pedestrian flow information to detect pedestrian crossing. To address this gap in the literature, this study employed pedestrian flow to detect pedestrian crossing patterns on the streets.

2.3 Pedestrian Crossing Detection

Pedestrian crossing information is highly relevant for autonomous vehicles. To increase the semantic information in a map [21] detected pedestrian crossing information using Bayesian classification techniques. However, this work only included 306 tracks, and the area used to detect pedestrians crossing was a crosswalk with an area less than $200m^2$. Moreover, this work only classified the tracks and failed to predict possible pedestrian movements beyond the captured movements. Alternatively, this study covered a street of approximately $1km$, with more than 2710 tracks. The pedestrian crossing was detected by using a pedestrian flow prediction algorithm that detected the crossing pattern of pedestrians on crosswalks as well as non-crosswalk regions.

In a few recent studies, i.e., [22–25], information from the environment was incorporated with the solutions to estimate the intentions of pedestrians. [26] used various contextual information, such as characteristics of the road,

presence of traffic signals, and zebra crossing lines, in conjunction with the pedestrians' state, to estimate if they planned to cross the road. They reported that using contextual information significantly improves the prediction of pedestrian crossing decisions. [27] predicted pedestrian crossing a street by elucidating pedestrians' relations, crosswalks, and ego vehicles, and combining two models: a standard inner-city model, always active, and a model activated only for crosswalks. We believe that our study is complementary to these previous works because we aim to build a map indicating the potential locations where pedestrians tend to cross the street. This map can be used as contextual information to support predictions of whether or not pedestrians cross the street. In this study, the prediction of the pedestrian crossing is performed by analyzing the pedestrian flow in the entire scene. Moreover, along with the dynamic information, we also employ semantic information, multiple daily data, and a graph approach to develop the pedestrian flow.

3 Information Extraction Through Mobile Vehicle Sensing

The trajectories of traffic participants are essential to understand the context of an environment. This section presents the difficulties associated with using trajectory information to extract information as well as methods to overcome these difficulties.

In the current study, an autonomous electrical vehicle (AEV) is used to acquire data. The AEV is equipped with many sensors, including a 360 degrees Velodyne HDL-32E LiDAR and a Ladybug5 360 degrees surround-view camera. The LiDAR sensor gives a 360 degrees point cloud of the surrounding area of the AEV with a 200 meters range, and the Ladybug generates five images that compose the surroundings of the AEV. Both camera and LiDAR acquire data at roughly 10 Hz. Although the LiDAR has a range of 200 meters, as the host vehicle drives within road speed limits during data acquisition, the pedestrian needs to be around 50 meters from the car to detect and track the pedestrian from the LiDAR point cloud. This is due to the sparse points cloud at the distant range. For this reason, we are provided with a short amount of time of pedestrian visibility, resulting in sparse pedestrian data. The authors propose to fuse multiple daily data to reduce this sparseness by detecting pedestrians in different areas. Furthermore, sparse data also contain mislabeled data. Thus, we employ the semantic data generated through the combination of LiDAR and camera data, which assigns a semantic label to each LiDAR scan point. Although a bounding box detection approach involving image data would be more efficient for detecting pedestrians, we chose the fusion approach because

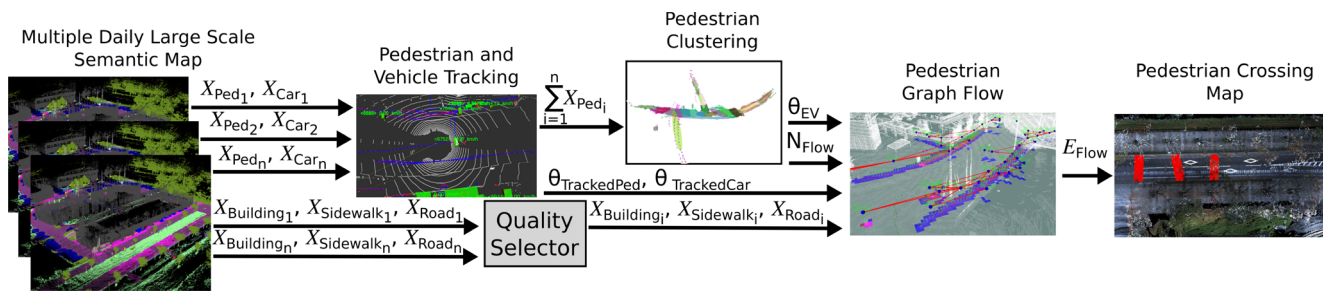


Fig. 2 Flow graph-based model representation

it accurately estimates pedestrians' position and yields additional information regarding surrounding environments, such as roads and sidewalks. To correct the semantic labels, we create a static map and use it to erase falsely labeled data.

Three main methods were used to predict traffic participants' flow: clustering of sparse trajectory data, representation of multi-direction for each cluster, and a set of heuristics for graph refinement. The process flow of this study is presented in Fig. 2. To develop the flow of pedestrians, multiple-daily large scale semantic maps were used to extract pedestrian (X_{Ped}), vehicle (X_{Car}), building ($X_{Building}$), sidewalk ($X_{Sidewalk}$) and road (X_{Road}) scan points [28]. Since $X_{Building}$, $X_{Sidewalk}$, and X_{Road} have similar information for every day's data, instead of fusing the multiple-day data, we choose to select the best day data based on the score of miss-classification for each of the labels, because fusing the data generates cumulative noise from the miss-classification of the labels. Subsequently, the pedestrian walking direction ($\theta_{TrackedPed}$) and vehicles driving direction ($\theta_{TrackedCar}$) are obtained by tracking the pedestrian and vehicle movement in the sequential LiDAR data. However, due to data sparseness, the tracking algorithms often fail to accurately track the objects, leaving many points without a heading direction. Furthermore, even with multiple daily data combined, the pedestrian flow is still sparse, and quite a few areas are left without any detected pedestrian. We employ a graph-based approach to overcome the sparseness. The combination of several days pedestrians points X_{Ped} are clustered generating the nodes (N_{Flow}), and a set of heuristics is employed to interconnect the clusters to generate the edges (E_{Flow}) of the flow graph. For the clustering, we chose an incremental Gaussian mixture model (GMM) clustering approach to perceive the pedestrian walking direction within the cluster's geometry (explained in detail in Section 4.3). This clustering method has the advantage that we can estimate the heading direction of the pedestrian through the Eigenvectors of the cluster (θ_{EV}), supplementing the heading directions estimated from the tracker $\theta_{TrackedPed}$. Moreover, the clusters are linked according to the multi-direction representation of pedestrian walking direction created for each cluster combining $\theta_{TrackedPed}$ and θ_{EV}

(explained in detail in Section 4.5). While creating the flow graph of the scene, one trade-off is between the coverage and the graph's accuracy. While choosing strict heuristics for the clusters' interconnection step would increase the accuracy, it would drastically limit its coverage. In this work, the authors choose to favor the coverage, which leads to many false positives edges within the flow graph initially. To compensate for this, false-positive edges are detected and erased using an obstacle cross (that uses the cleaner $X_{Building}$), road cross, and redundant connection removal algorithm. Finally, pedestrian road crossing information is obtained by analyzing the edges (E_{Flow}) of the pedestrian flow with the cleaner X_{Road} and $X_{Sidewalk}$ selected among the multiple-daily data.

4 Methodology

4.1 Cleaning of Semantics Data

Semantic information was extracted for each scan point by fusing the camera and LiDAR data [28]. Westfechtel et al. reported an average Intersection over Union (IoU) score of 65% for all labels and an IoU score of 84% for pedestrians. To further increase the robustness of results, filtering techniques are introduced in this section. We observed several miss-classifications in the resulting semantic map; however, on further examining the semantic map, we found that the occurrence of miss-classification tended to become more sparse (i.e., only a few points of a vehicle were miss-classified as buildings). Therefore, we chose statistical outlier removal (SOR) [29] to filter out these miss-classifications from the buildings, roads, and sidewalks. After, a grid map is created for each class. The building grid map is used for graph refinement (Section 4.6), and sidewalk and road grid maps are used to identify pedestrian crossings (Section 4.7).

Subsequently, for pedestrians and vehicles, the miss-labeled points were often attributed to a small-time gap between the LiDAR data and camera data acquisition, which leads to a miss-projection between the two modalities. This miss-projection results in the mislabeling of several

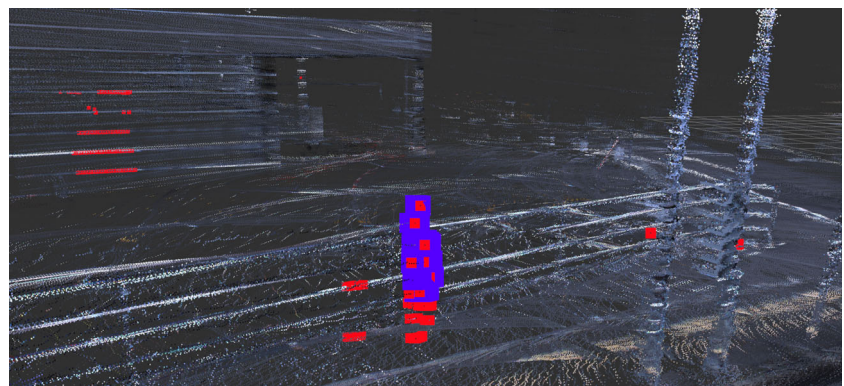
points in one location; thus, SOR was not suitable for removing these outliers. Our approach exploits that the miss-projections are projected onto other objects of the surrounding. While the number of mistakenly projected semantics is not low enough for the SOR approach, looking at the object, the correctly projected semantics (i.e., building, vegetation, among others) outweigh the incorrect projections. Therefore, a static map (X_{map}) is created by removing all the points related to dynamic objects such as pedestrians, vehicles, and bikes. Then, the minimal distance of each pedestrian scan point to the point of the static map X_{map} is calculated, as follows:

$$d = \min_{i \in N} \|X_{Ped} - X_{map}^i\| \tag{1}$$

$$\begin{cases} X_{Ped} \in X_{conf}, & \text{if } d \geq d_{threshold} \\ X_{Ped} \notin X_{conf}, & \text{otherwise} \end{cases} \tag{2}$$

where d is the minimum Euclidean distance between a pedestrian point and all the points in X_{map} , $d_{threshold}$ is a distance threshold of 0.25 m, and X_{conf} is the resulting points obtained after map filtering. Even after eliminating the points that are closer to the static map objects, a few points remain at several places on the map. To resolve this, an Euclidean cluster is used for each single scan of the LiDAR with a minimum cluster value of 50 points. Figure 3 shows the pedestrian points after the proposed filtering in blue. The points that were removed during this process are shown in red.

Fig. 3 Pedestrian scan point cleaner example: **a** The pedestrian points obtained after the proposed filtering in blue. The points that were removed during this process are indicated in red. The process is done for each single LiDAR scan, while the semantic map contains points of the whole run. **b** Actual image of the scan points



(a)



(b)

4.2 Trajectory Estimation from Pedestrian Point Cloud

As we are using a mobile probe to acquire the data in this work, single pedestrians are only tracked over a short amount of time. Therefore, the trajectories are scarce, and pedestrian detections that could not be tracked would not be considered. Therefore, we chose a different approach. We estimate the pedestrian flow by clustering the individual pedestrian points and interconnecting these clusters based on the walking direction of the pedestrians and the geometric features of each cluster. To estimate the pedestrians’ walking direction, we employ the IMM-UKF-PDA Tracker proposed by [30] to track pedestrians in the sequential LiDAR scans of the single daily data. We extract the walking directions ($\theta_{PedTracked}$ and $\theta_{CarTracked}$) of the tracked objects and append this information to the respective cluster of pedestrians, X_{Ped} for pedestrians and X_{Car} for vehicles respectively. However, as some of the X_{Ped} could not be tracked, we further estimate the pedestrian’s walking direction by analyzing each pedestrian cluster’s geometric properties using the principal component analysis (PCA).

4.3 Clustering of Sparse Trajectory Data

For our pedestrian flow estimation approach, we need to cluster the pedestrian points X_{Ped} first. The clustering should ideally cluster pedestrian points that are walking in the same direction and are locally close. One of such desired cluster

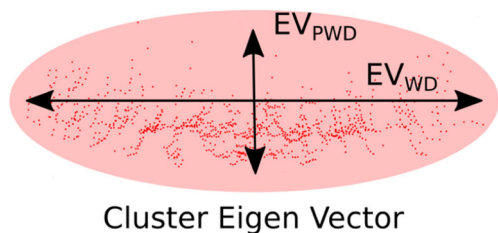


Fig. 4 Cluster of pedestrian showing the Eigenvector of the walking direction (EV_{WD}) and Eigenvector perpendicular to the walking direction (EV_{PWD})

can be seen in Fig. 4. It can be seen that the cluster is large within the walking direction but is limited in the direction perpendicular to it. In order to exploit this, we chose an incremental GMM clustering approach. The GMM clustering uses 2-dimensional Gaussian curves to represent each cluster. The Gaussian curves' variances can be interpreted as the cluster's extension in the walking direction and the perpendicular direction. However, for the conventional GMM clustering method, the amount of desired clusters has to be given as input, which is not applicable in our case. We, therefore, implemented an incremental GMM clustering approach. We run the GMM clustering for one cluster at first. We analyze the variance of the GMM perpendicular of the walking direction EV_{PWD} . If this variance is above a certain threshold, we assume that different walking directions are present within the estimated cluster. We, therefore, increase the amount of desired clusters by one and rerun the clustering process. This process increases the number of desired clusters until each of the estimated clusters has a variance for the direction perpendicular to the walking direction EV_{PWD} below our set threshold. We chose a threshold of 1 m in our implementation. Figure 5 shows four iterations of the GMM clustering in one cluster.

Due to our clustering approach's iterative nature, the clustering cannot cluster the whole pedestrian points X_{ped} at once. Therefore a DBScan clustering algorithm is used as a pre-clustering step to cluster all points with proximity. To improve the system, a minimum cluster size of 50 points was chosen to erase noisy points.

Figure 6 presents the trade-off between our approach and mean shift clustering. The corresponding angle of

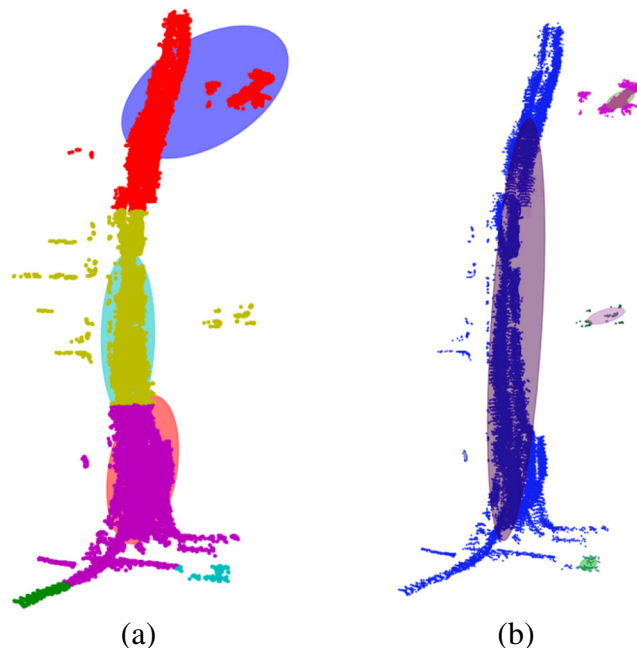


Fig. 6 Data clustered using: **a** mean shift; and **b** proposed approach. The different point colors correspond to different point clusters, and the ellipses represent the Eigen vectors of the cluster

EV_{WD} of the cluster (θ_{EV}) is used to improve the estimation of the pedestrian flow direction. The mean shift approach shows a less accurate correlation between the direction of pedestrians movement and EV_{WD} . For example, Fig. 6a presents a red cluster of pedestrian scan points, which moved from the south to the north; however, EV_{WD} indicates that these points moved from southeast to northeast. This discrepancy was attributed to the incorrect clustering. On the contrary, the GMM approach features better similarity between EV_{WD} and the direction of pedestrian movement, as shown in Fig. 6b.

4.4 Fusion of Multiple Daily Data

As previously mentioned, one challenge of this study is data sparseness. To overcome this, we employ two main strategies; fusing multiple daily data and predicting connections between pedestrian clusters, generating a flow graph. As the graph-constructing algorithm infers

Fig. 5 Figure shows the GMM cluster iterations of one cluster. The first image represents the first iteration of the GMM on the cluster, the third image represents the second iteration of the cluster, the third image represents the 15th iteration of the GMM in the cluster, and the last image represents the last iteration of the GMM in the cluster

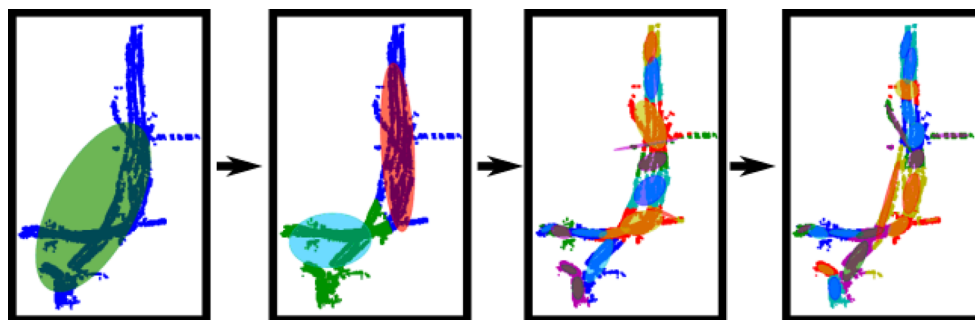
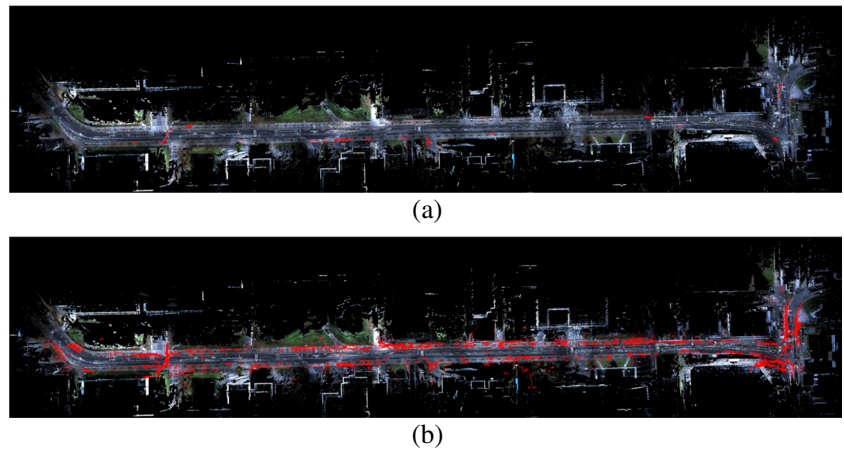


Fig. 7 Coverage of pedestrians (in red) on the map: **a** 1 day data; **b** 11 days data



connections based on existing clusters, the inference increases with the number of pedestrian clusters analyzed. We, therefore, chose to fuse multiple daily data before the graph estimation. In particular, we fuse the data before the clustering step, the trajectories ($\theta_{PedTracked}$ and $\theta_{CarTracked}$) for the points are estimated for each trial separately. To fuse the data, we align each daily data using the normal distributions transform (NDT) matching algorithm [31]. Once the point clouds are aligned, we simply accumulate the pedestrian X_{Ped} . Figure 7 depicts (a) data of one trial and (b) data after fusing 11 trials; where the pedestrian scan points (red dots) increased when using the multiple daily data.

4.5 Estimation of Pedestrian Flow for Each Cluster

In the construction of the flow graph, the pedestrian flow of each cluster is one main component to interfere connections between the clusters. For each cluster, the set of pedestrian tracked direction ($\theta_{PedTracked}$) within the cluster is associated to the cluster. Moreover the pedestrian flow can also be estimated through the geometry of the cluster (θ_{EV}).

We fuse these information using a Gaussian Mixture Model (GMM):

$$P(\theta) = \frac{1}{\kappa} \cdot \sum_{i=0}^n \frac{1}{\sigma\sqrt{2\pi}} e^{-(\theta-\theta_{PedTracked})^2/2\sigma^2} + \frac{1}{\sigma\sqrt{2\pi}} e^{-(\theta-\theta_{EV})^2/2\sigma^2} \quad (3)$$

Where n is the number of associated $\theta_{PedTracked}$ in the cluster, and $P(\theta)$ corresponds to the probability of a pedestrian moving in the direction of the cluster. The normalization κ was chosen so that it is the maximum value of the sum of Gaussians:

$$\max_{\theta} \frac{1}{\kappa} \cdot \sum_{i=0}^n \frac{1}{\sigma\sqrt{2\pi}} e^{-(\theta_{PedTracked}-\mu)^2/2\sigma^2} = 1 \quad (4)$$

To decide whether the cluster has a direct flow in a specific direction, we employ a direction threshold P_{Thresh} . Only angles for which $P(\hat{\theta}) > P_{Thresh}$ are considered as flow directions for the cluster. It has to be noted that this representation allows each cluster to have a pedestrian flow in multiple directions, as can also be seen in Figs. 8 and 9.

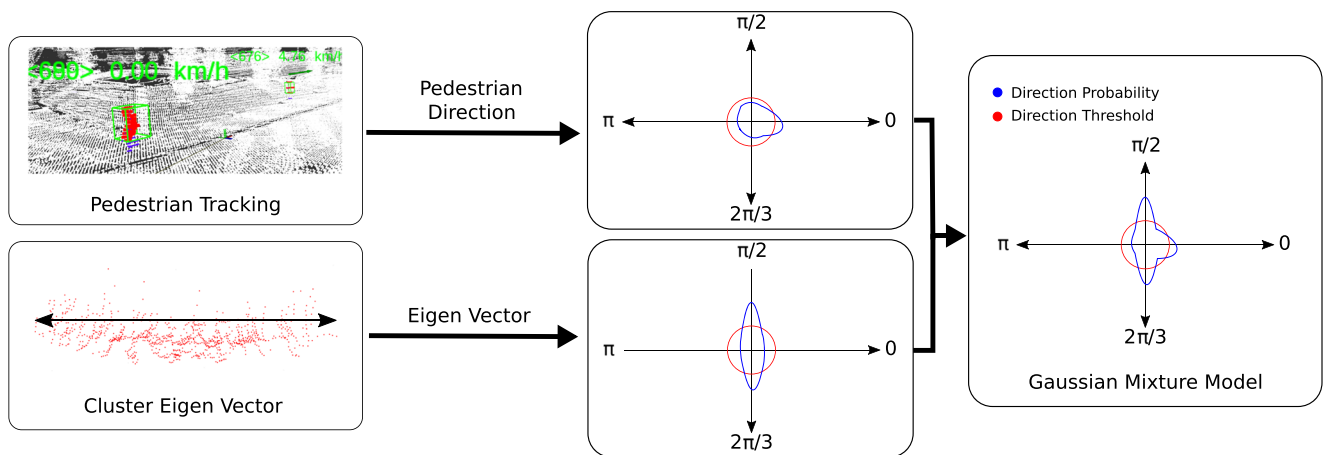
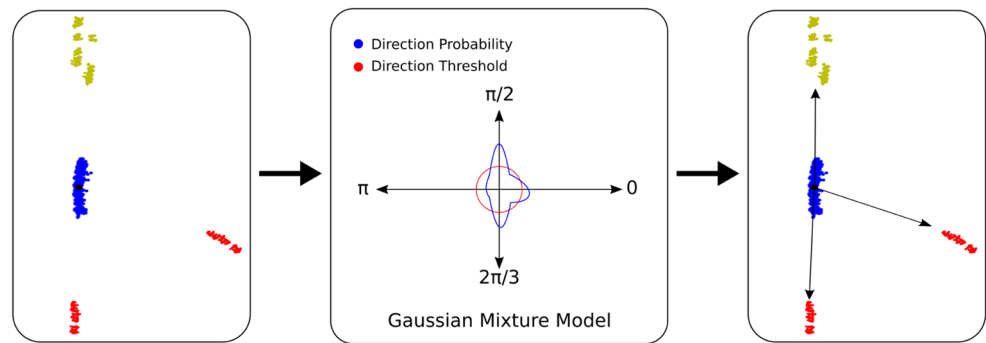


Fig. 8 Overview of GMM for edge estimation

Fig. 9 Example of edge estimation



4.6 Graph Refinement

In this work, the authors choose to favor the coverage, which leads to many false positive edges within the flow graph (E_{Flow}) initially. To compensate for this, false-positive edges are detected and erased using an obstacle cross, road cross, and redundant connection removal algorithm. We chose those methods aiming to remove the false-positive edges without reducing the pedestrian flow information. In Fig. 10 some examples of false positive edges can be seen.

On analyzing the graph, it is evident that some of the edges pass through buildings in the map (green edges), and some edges cross the street diagonally to the other side of the street, which is an irregular path for pedestrians (red

edges). Hence, these edges were removed to create a more accurate graph of pedestrian flow. After removing the first two cases, it was observed that one node was connected to several other nodes in the same direction. To obtain a clean representation of pedestrian flow, each node in the graph should be connected to the closest node along each direction. Therefore, we removed redundant edges (yellow edges) from the graph. The procedure to detect each false positive edge is explained in the following sections.

4.6.1 Obstacle Cross Outlier

When estimating pedestrians' flow, buildings are considered obstacles which do not allow pedestrians to pass. Therefore,

Fig. 10 Overview of graph refinement indicating the edges before and after graph refinement, where each refinement process is illustrated. The green edges correspond to building cross outliers, red edges correspond to road cross outliers, and yellow edges correspond to redundant connection outliers

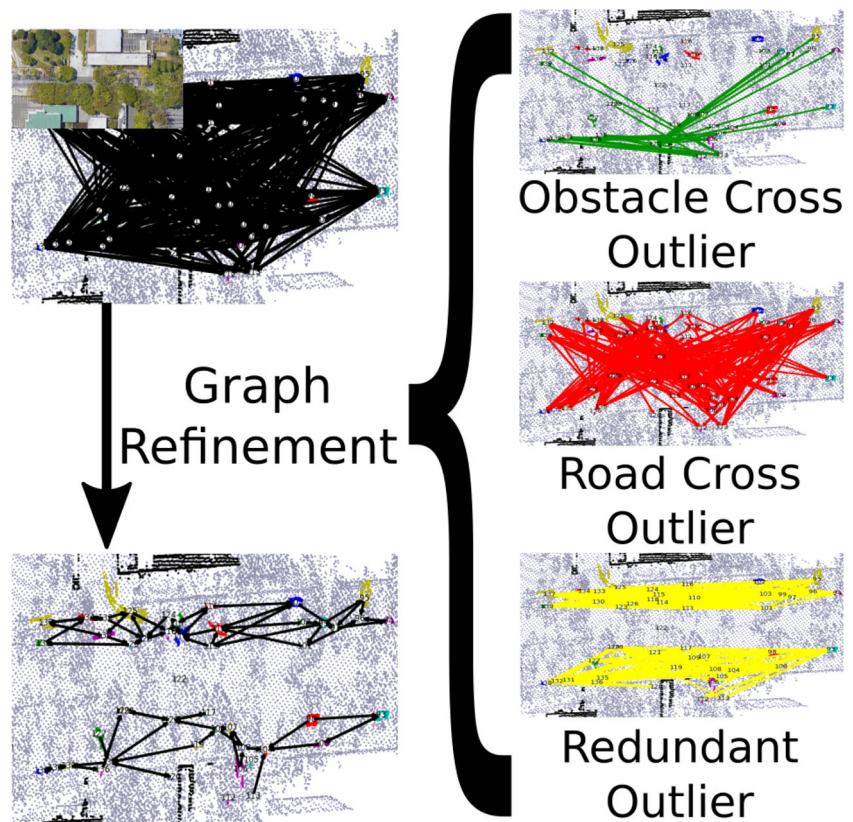
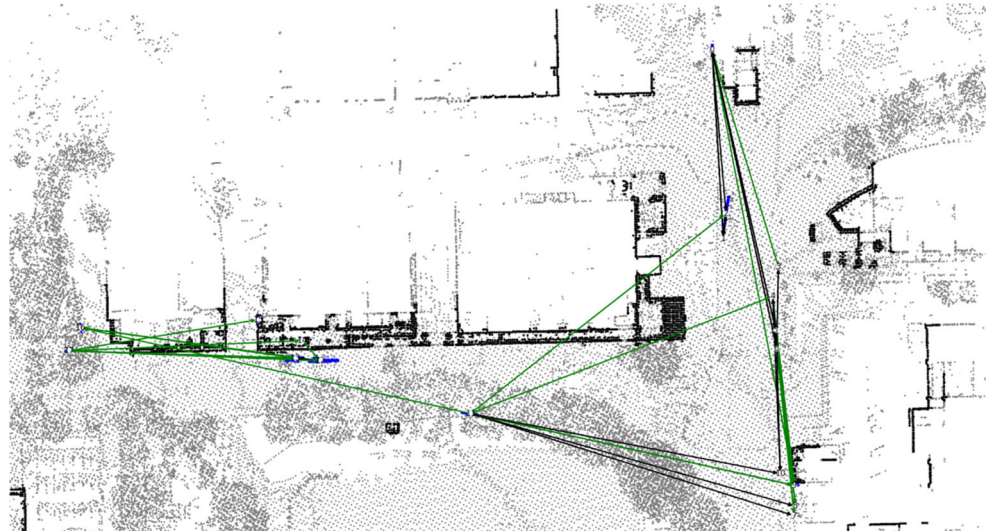


Fig. 11 An edge outlier of building crossing. The black dots correspond to buildings, blue dots correspond to pedestrians, green lines are the edge outliers and black lines are the appropriate links



we consider any edge predicting a pedestrian flow through a building as a false positive that should be removed. To determine if an edge crosses such an obstacle, we employ the building map $Grid_{Building}$ (see subsec. 4.1). Each edge is tracked along its length and evaluated whether it crosses an obstacle grid map generated from $X_{Building}$. If the algorithm detects a conflict between the grid map and the edge, the edge is labeled green and removed. Figure 11 highlights such green edges crossing buildings.

4.6.2 Road Cross Outlier

In this work, we presume that pedestrians cross the street in a straightforward way, which would be to cross the street perpendicular. Under this assumption, we label each graph edge that crosses the street with an angle not perpendicular, with a 30° margin ($\theta_{CrossThreshold}$), as a false positive edge.

We choose the margin of 30° because the angle is estimated by employing the trajectories of pedestrians and vehicles (including the mobile probe) each with 15° of noise.

$$\begin{cases} E_{Flow} \in E_{CarCross}, & \text{if } |\angle(E_{Flow}, \theta_{CarTraked}) - 90^\circ| < \theta_{CrossThreshold} \\ E_{Flow} \in E_{CrossOutlier}, & \text{otherwise} \end{cases} \quad (5)$$

where \angle gives the angle of intersection between the graph edge E_{Flow} and $\theta_{CarTraked}$, $E_{CarCross}$ corresponds to the edges that crossed a vehicle's trajectory, and $E_{CrossOutlier}$ corresponds to the edges that cross the street in an irregular trajectory and are eliminated. The Fig. 12 presents a few cases where the where the edges cross the path of the vehicle. On the upper left and the lower right sides, the edges exceed the threshold.

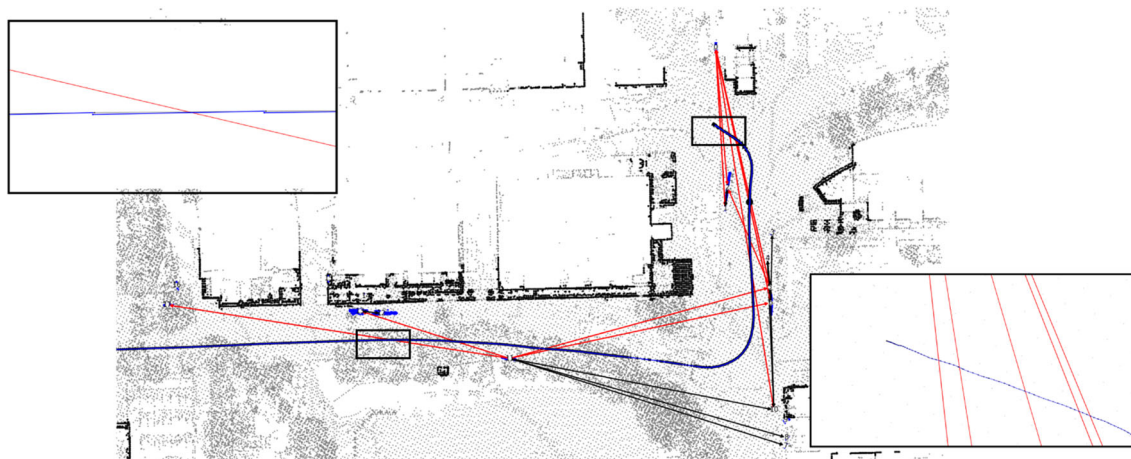
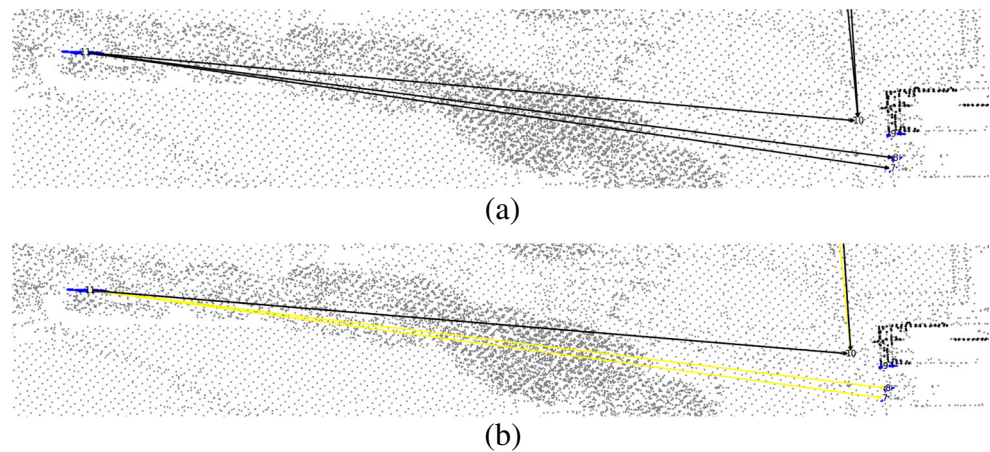


Fig. 12 An edge outlier for crossing the street. The blue edges represent the vehicle's trajectories, red dots are the edge outlier, and black edges are appropriate links

Fig. 13 Example of edge density reduction. **a** Before; **b** After



4.6.3 Redundant Connections Outlier

This heuristic aims to limit the connections of nodes along the same direction, in order to clean the graph. As shown in Fig. 10, a few nodes in the graph contain several connections along the same direction. To eliminate such redundant connections, each node connection is checked; if the node has more than one connection along the same direction, only the shortest connection is retained, and the remaining connections are eliminated from the graph. Equation 6 describes the process used to determine redundant outliers:

$$\begin{cases} E_{SameDirection}^{i,j} = f(E_{Flow}^i, \theta_{DirectionThreshold}) \\ E_{RedundantOutlier} = \sum_{i=0}^n \sum_{j=0}^m (\min_{dist}(E_{SameDirection}^{i,j}) - E_{SameDirection}^{i,j}) \end{cases} \quad (6)$$

where E_{Flow}^i are the edges of the node i , $\theta_{DirectionThreshold}$ is the direction threshold of 25° , $E_{SameDirection}^{i,j}$ are the j set of edges along the same direction in the node i , f corresponds to the function that checks if the edges are heading along the same direction, \min_{dist} is the function that returns the edge with the minimum Euclidean distance, n is the total number of nodes, m is the number of directions with multiple edges for each node i , and $E_{RedundantOutlier}$ are redundant edges that are to be removed. Figure 13 presents an example of such redundant connections.

4.7 Pedestrian Crossing Detection

It is possible to automatically extract pedestrian crossing patterns on a road, by analysing a pedestrian flow graph. Based on the set of edges $E_{CarCross}$, there are two types of edges: edges where pedestrians crossed the path of a car on road ($E_{RoadCross}$), and edges where pedestrians crossed the path of a car on the sidewalk ($E_{SidewalkCross}$), which some

represent entry driveways. Semantic information of the road is used to distinguish the $E_{RoadCross}$ edges from $E_{CarCross}$ edges. Equation 7 demonstrates the selection of $E_{RoadCross}$:

$$\begin{cases} E_{CarCross} \in E_{RoadCross}, & \text{if } E_{CarCross} \subset Grid_{Road} \\ E_{CarCross} \in E_{SidewalkCross}, & \text{otherwise} \end{cases} \quad (7)$$

where $Grid_{Road}$ is the grid map of the road points extracted from the semantic map. Figure 14 presents an example of $E_{RoadCross}$ in purple and $E_{SidewalkCross}$ in orange.

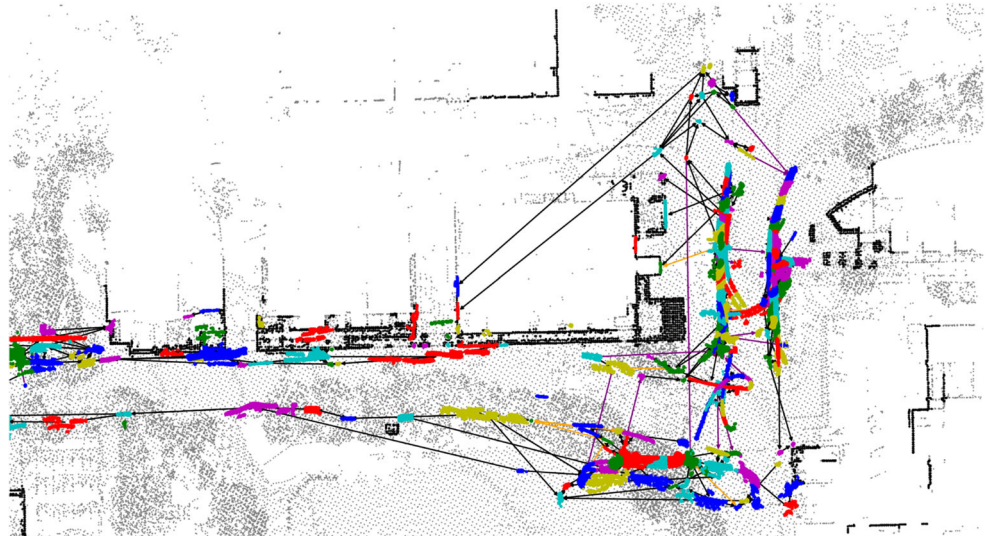
5 Evaluation

5.1 Data and Environment

5.1.1 Mobile Probe Data

An AEV was used to gather data for two environments of Sendai, Japan. The AEV is equipped with a multitude of sensors, including a Velodyne HDL-32E Lidar and a Ladybug5 surround-view camera. Throughout experiment A, the AEV gathered LiDAR and camera data. For the experiment A, the AEV drove along a course of approximately 1 km for a total of 11 times while extracting the LiDAR scan points and camera images for this research on a Campus area. The pedestrian data was captured during fall, winter, and spring, for two days, one day, and eight days, respectively. Four of those days were Monday, six days were Tuesday, and one day was Friday. For experiment B, The AEV drove along a course of approximately 1.5 km a total of 3 times in the City area. The pedestrian data was captured during the fall season for three days. One of those days was Sunday, another day was Tuesday, and the last day was Thursday. Figure 15 depicts the AEV in this study.

Fig. 14 2 days data flow of pedestrian showing the $E_{CarCross}$ edges separated in purple edges ($E_{RoadCross}$) and orange edges ($E_{SidewalkCross}$)



5.1.2 Experiment Environment

The map was built using simultaneous localization, and mapping [32] of the LiDAR data obtained from the mobile vehicle sensor. After fusing the LiDAR and camera data, each point of the map contained color and semantic information. Figure 16 was obtained by extracting the sidewalk points.

5.2 Evaluation of Graph Refinement

5.2.1 Area of Evaluation

All the regions from experiment A and B were used to evaluate the accuracy of the graph refinement heuristics. Small areas of the environment, corresponding to 200 m and 100 m of the vehicle course from experiment A and B, respectively, were used to illustrate the refinement’s final process. Figure 17 presents the map used for this evaluation, as well as the small squares on the map delimiting the

regions selected for illustration. To create the ground truth for this evaluation, each edge of the graph is manually counted and checked for each heuristic of graph refinement explained in Section 4.6.

5.2.2 Accuracy of Heuristics

To determine the accuracy of heuristics used in this study, the following equation is used for each heuristic:

$$Acc_{Edges} = \frac{E_{DetectedHeuristic} \cap E_{HeuristicGroundTruth}}{E_{HeuristicGroundTruth}} \quad (8)$$

where Acc_{Edges} corresponds to the accuracy of edges that are detected in the heuristic, $E_{DetectedHeuristic}$ is the number of edges that were detected for the heuristic, and $E_{HeuristicGroundTruth}$ is the total number of edges manually counted for the heuristic.

5.3 Evaluation of Cover Ratio

In this section, we evaluate the cover ratio of the sidewalk. We assume that all of the sidewalks in the surrounding area are used by pedestrians. As it is hard to create ground truth data, we chose the sidewalk to evaluate pedestrian coverage. Our reasoning for choosing the sidewalk as a parameter for pedestrian coverage is that if we could see the entire movement of pedestrians along the evaluated region at any given moment in time, most likely it would show that the place that pedestrians use the most—in other words, the sidewalk—is being completely covered. While pedestrians do not necessarily use the entire sidewalk area, we believe it is a good enough approximation as most of it is commonly used. By comparing the sidewalk area extracted from the semantic map with the pedestrian points and the flow graph,

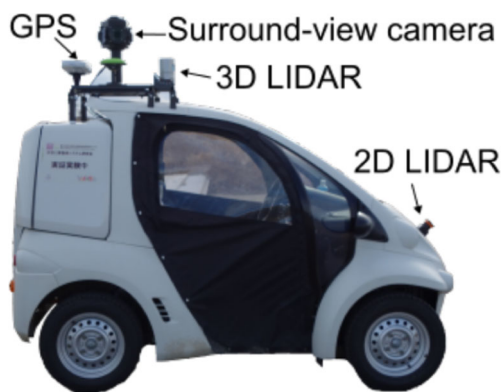
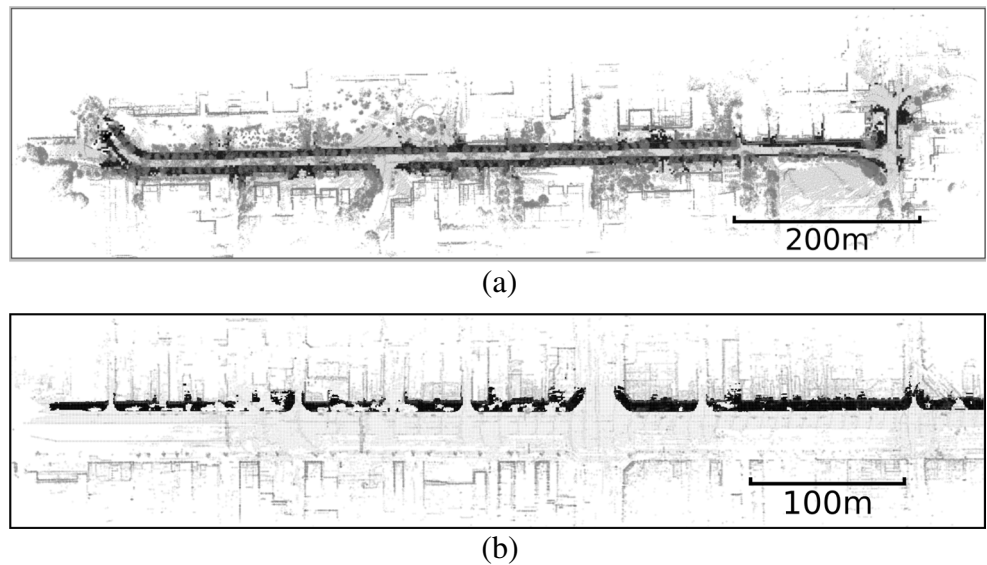


Fig. 15 Autonomous electric vehicle used in data acquisition experiments

Fig. 16 Sidewalk scan points extracted from the semantic map, grid map in black with gray dots representing the static map: **a** Experiment area A; **b** Experiment area B



we can analyze whether our research methods have any impact on the cover ratio of the sidewalk.

5.3.1 Evaluation Method

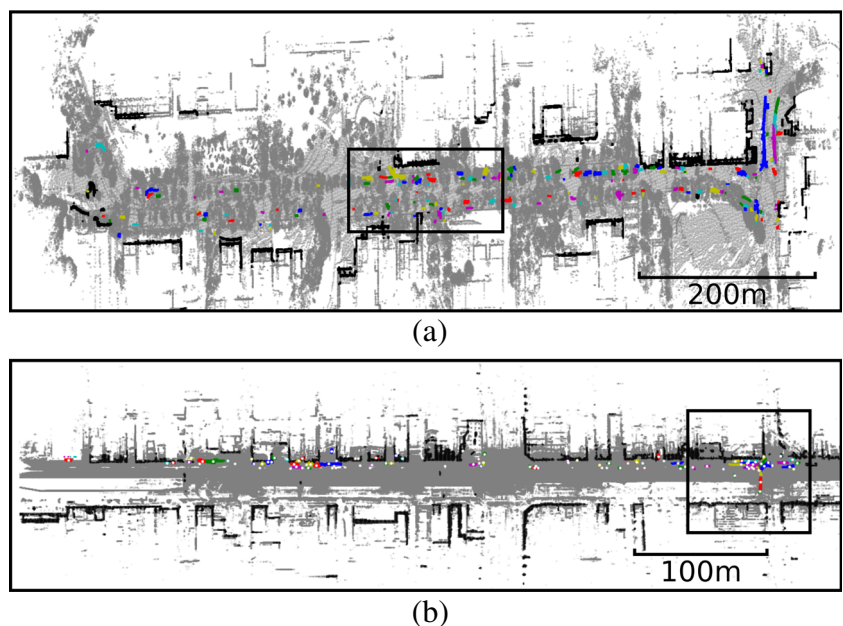
In order to calculate the cover ratio of the sidewalk, we employ grid maps with a cell size of 1m, resulting in grid maps of roughly a width of 1000 cells and a length of 400 cells. For experiment A, there are 11 days data. For this evaluation, each day will consist of a combination of the days' data. For example, two days data combination consists in the combination of days 1 and 2, 2 and 3, 3 and 4, and so on until the 10th and 11th combination. For each combination of days, a mean and standard deviation

are computed. We calculate the cover ratio as the number of occupied grid cells of the intersection between the pedestrian and the sidewalk divided by the number of occupied cells of the sidewalk map:

$$Rat_{pedestrian} = \frac{|G_{pedestrian,occ} \cap G_{sidewalk,occ}|}{|G_{sidewalk,occ}|} \quad (9)$$

where $Rat_{pedestrian}$ corresponds to the cover ratio, $|G_{sidewalk,occ}|$ corresponds to the amount of sidewalk grid cells and $|G_{pedestrian,occ} \cap G_{sidewalk,occ}|$ corresponds to the number of cells of the sidewalk where pedestrians were detected. Figure 16 presents the sidewalk grid map.

Fig. 17 Graph refinement evaluation experiment area and results illustration area on squares. Pedestrian points are marked in different colors per cluster and the buildings points in black: **a** Experiment A; **b** Experiment B



5.4 Pedestrian Cross Evaluation

5.4.1 Area of Evaluation

For the evaluation of pedestrian crossing, the entire map of the environment is used, as shown in Fig. 17. To obtain the ground truth of the evaluation, pedestrian movement in the entire map was observed. Thereafter, the areas where pedestrians tend to cross the street and the areas where crosswalks can be seen were manually drawn on the grid map, as shown in Fig. 18.

5.4.2 Performance Evaluation

To evaluate pedestrian crossing detection performance, a ground truth map of the observed pedestrian crossing regions was built manually. For experiment A, this ground truth map corresponded to 13 locations containing crosswalks and 9 locations where pedestrians were observed to cross the street without using crosswalks. The regions without crosswalks were observed with and without the use of the probe vehicle, and they primarily consist of locations in proximity to entrances of buildings. For experiment B, this ground truth map corresponded to 10 locations containing crosswalks. For this evaluation, the percentage of the manually observed crossing regions detected by our algorithm is computed as follows:

$$Reg_{recall} = \frac{|Reg_{detected} \cap Reg_{spotted}|}{|Reg_{spotted}|} \quad (10)$$

where Reg_{recall} corresponds to the percentage of the regions detected using the proposed algorithm, $|Reg_{spotted}|$ denotes the total number of regions manually spotted by the

authors and $|Reg_{detected} \cap Reg_{spotted}|$ the number of regions of the ground truth data that our algorithm succeeded to identify.

6 Results

6.1 Graph Refinement

For experiment A, the result of the flow graph estimation containing all 2794 edges was computed for the whole experiment area; 87% of the refinement showed false positive edges. Based on the false-positive edges, 242 edges (10%) corresponded to the building crossing outlier, 971 edges (40%) corresponded to the street crossing outlier, and 1231 edges (50%) corresponded to redundant outliers. For experiment B, the flow graph estimation result containing all 3008 edges was computed for the whole experiment area; 96% of the refinement showed false positive edges. Based on the false-positive edges, 648 edges (23%) corresponded to building crossing outlier, 739 edges (25%) corresponded to the street crossing outlier, and 1521 edges (52%) corresponded to redundant outliers. Figure 19 shows the result of the graph refinement process in the regions of experiment A and B.

6.2 Cover Ratio

For experiment A, Fig. 20 shows the cover ratio of the sidewalk using the pedestrian points before creating the flow graph, and the cover ratio after estimating the pedestrian flow employing the different amount of daily data. It can be seen that estimating the pedestrian flow increases the cover ratio by an average of about 14 percentage points compared

Fig. 18 Grid map of regions where pedestrians were spotted crossing the street. Where the green areas represent the regions observed by the authors, and the blue areas represent crosswalk regions: **a** Experiment A; **b** Experiment B

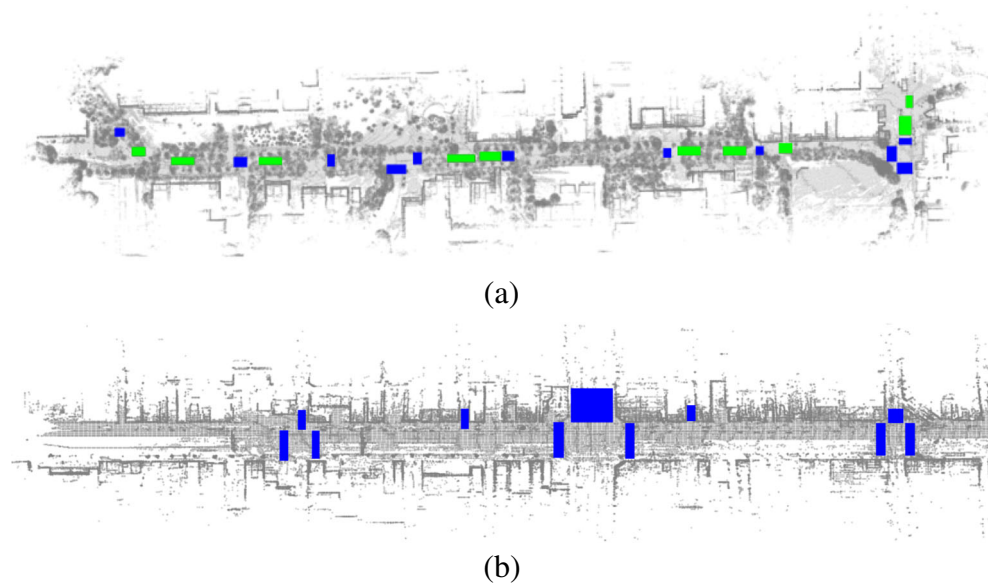
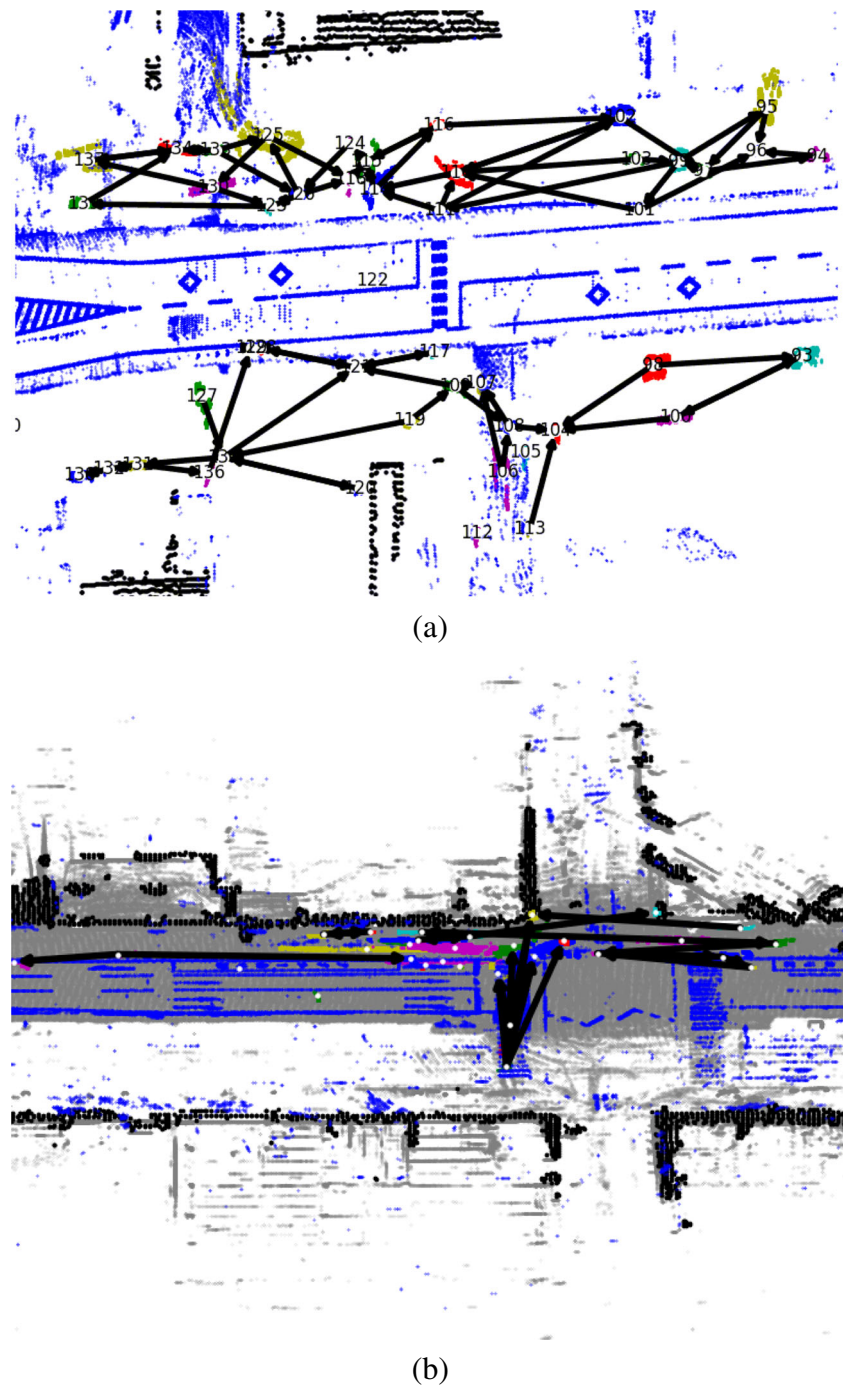


Fig. 19 Region of the map indicating the edges remaining after removal of outliers during the graph refinement step. The black edges correspond to pedestrian flow directions, blue dots correspond to traffic environment, and black dots are buildings: **a** Experiment A; **b** Experiment B



to only using pedestrian points. The use of multiple data and pedestrian flow approaches has increased pedestrians' cover rate by around 61% points, resulting in a final cover rate of 69% of the sidewalk. Furthermore, we can significantly increase the cover ratio through the combination of multiple daily data. Figure 21 shows the grid map of the coverage of pedestrians' flow on the sidewalk.

For experiment B, the cover ratio of pedestrians consisted of 12%, whereas the pedestrian flow corresponded to 24%. It can be seen that estimating the pedestrian flow increases

the cover ratio by an average of about 12 percentage points compared to only using pedestrian points. Figure 22 shows the grid map of the coverage of the flow of pedestrians on the sidewalk for experiment B.

6.3 Detection of Pedestrian Crossing Pattern

For the experiment A, based on the 21 regions spotted by the authors ($Reg_{spotted}$), 17 regions were successfully detected by the proposed algorithm, resulting in a Reg_{recall}

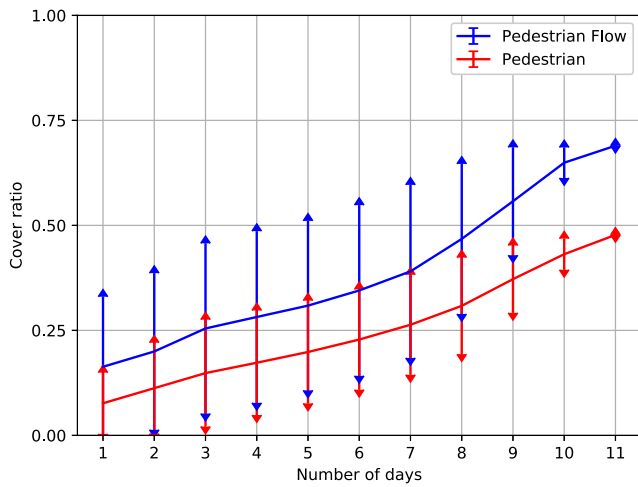


Fig. 20 Ratio of sidewalk coverage by the number of days data from 0 to 1, where the red line correspond to the pedestrian scan points, and blue line correspond to the graph flow of pedestrians

of 81%. Among the four regions that were not detected, three were crosswalk regions. For experiment B, based on the 11 regions spotted by the authors ($Reg_{spotted}$), six regions were successfully detected by the proposed algorithm, resulting in a Reg_{recall} of 55%. The five regions

that were not detected were crosswalk regions. Figure 23 shows the directed graph representing the pedestrian flow.

Furthermore, about 25% of the crossing patterns detected by our algorithm did not correspond to regions manually observed by the authors. These crossing patterns are marked in red in the Fig. 23. Moreover, Fig. 24 shows places detected by the algorithm in an RGB map of the environment. Where the green areas represent the regions observed by the authors, the blue areas represent crosswalk regions, and the red areas represent new regions detected by the algorithm

7 Discussion

The results encompassed two experiments, each with different features. Experiment A represents a campus area of dual-lane traffic of two hands of low traffic movement, where the data was captured on a lapse of 11 days. Experiment B represents a city area of one hand traffic on three lanes of high traffic movement, where the data was captured on a lapse of 3 days.

The proposed algorithm was able to estimate the pedestrian flow from 69% and 24% of the sidewalk area

Fig. 21 Grid map of experiment A: **a** Sidewalk; **b** Pedestrian scan points on sidewalk; **c** Pedestrian Flow graph on the sidewalk

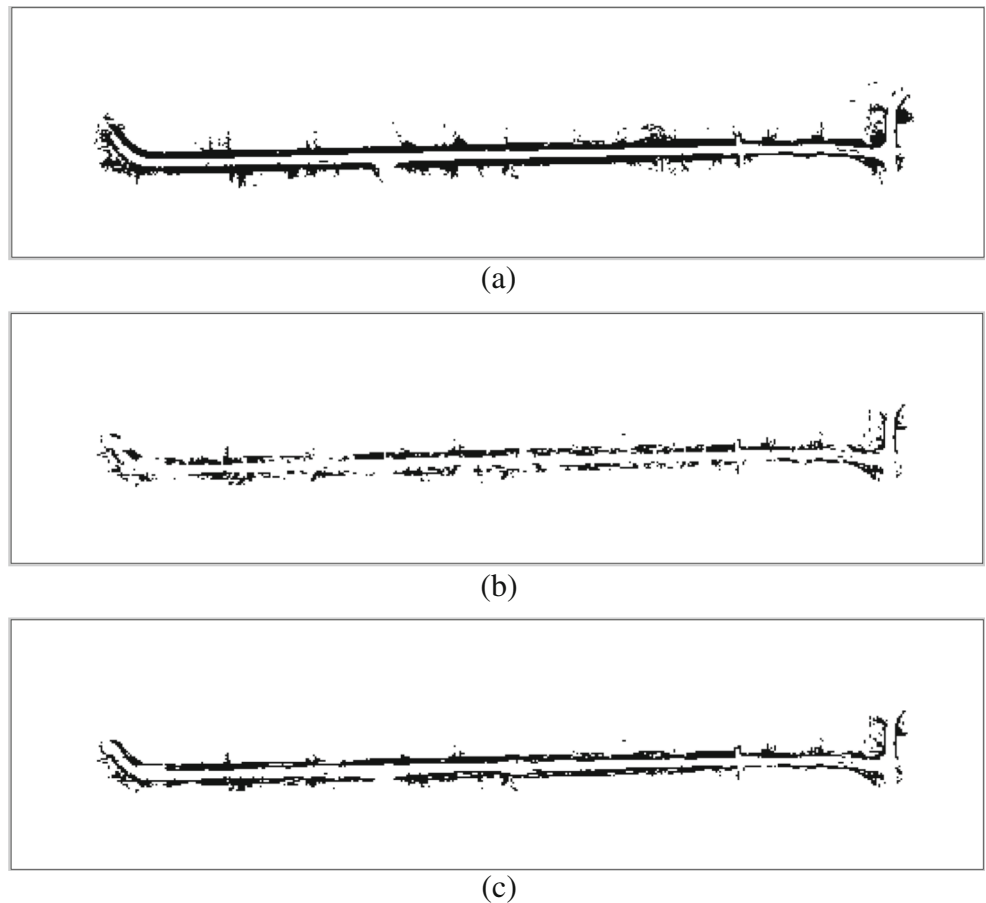
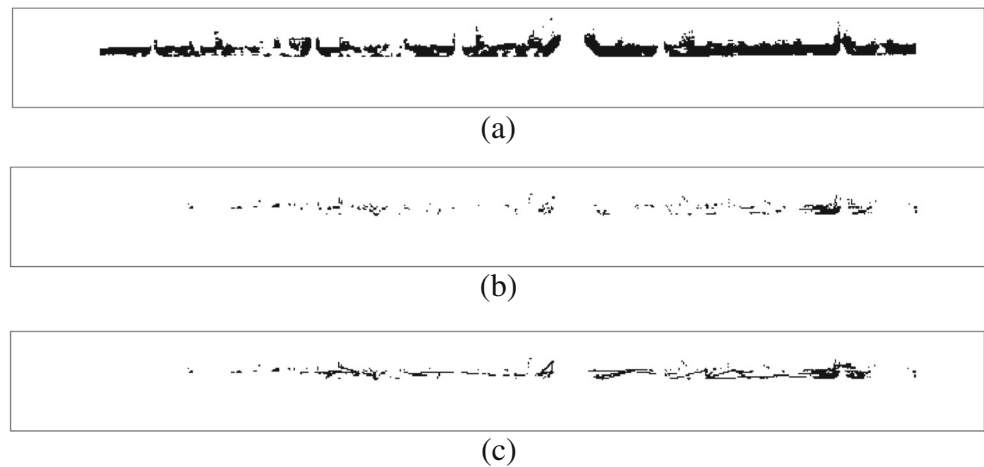


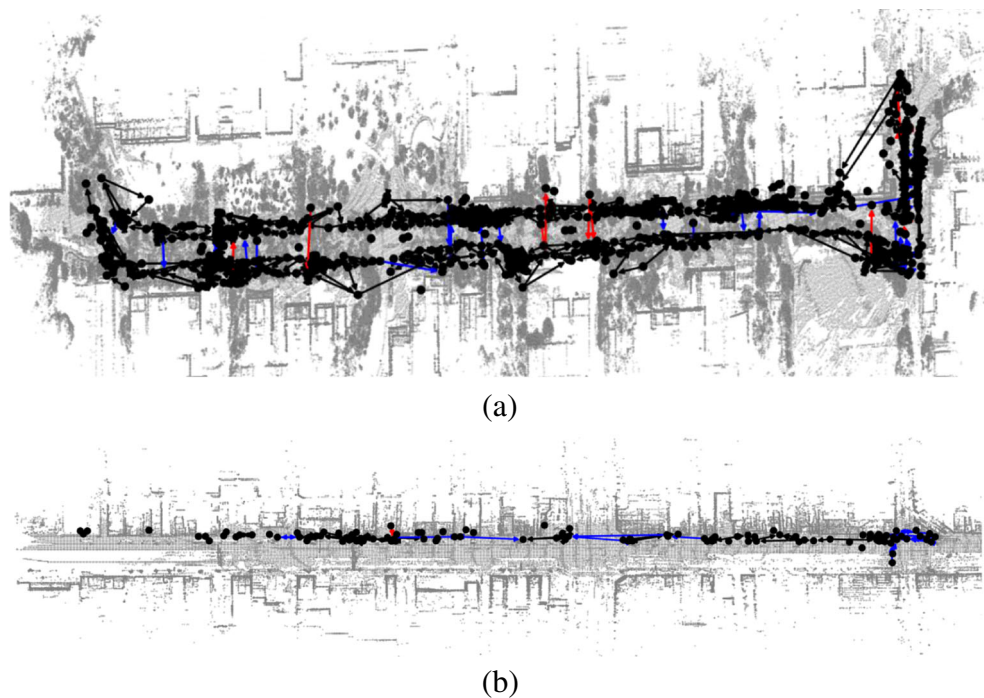
Fig. 22 Grid map of experiment B: **a** Sidewalk; **b** Pedestrian scan points on sidewalk; **c** Pedestrian Flow graph on the sidewalk



for the experiment A and B, respectively, using multiple daily sparse data of pedestrians, through the GMM-based clustering and graph-based approaches. In the case of experiment A, the use of multiple daily data increased the density of pedestrians from 8% to 48%. Moreover, the density of graph-based pedestrian flow increased from 16% to 69% of sidewalk coverage. Although pedestrians' flow reached 69%, as shown in Fig. 21, the pedestrian flow obtained covers most of the sidewalk. The low percentage could be explained as pedestrians usually stick to the center part of the sidewalks rather than the border areas. Therefore often, the whole width of the sidewalk is not estimated by our algorithm.

The outlier techniques described in this work successfully detected and eliminated 87% of edges from the graph for experiment A and 96% of edges from the graph for experiment B, which indicates the importance of graph refinement. The set of heuristics created herein aimed to achieve a more concise and accurate pedestrian flow by eliminating the connections that are unlikely to be executed by pedestrians. Based on experiment area A and experiment area B used for this evaluation, it was evident that all the removed edges corresponded to the respective heuristics. By comparing the graph refinement from experiment A to B, one can see a decrease in the percentage of road cross edges on experiment B due to a lower detection of

Fig. 23 Directed graph indicating pedestrian flow. The edges in blue correspond to the crossing patterns detected in regions spotted by the authors, and the edges in red correspond to crossing patterns in regions not spotted by the authors: **a** Experiment A; **b** Experiment B



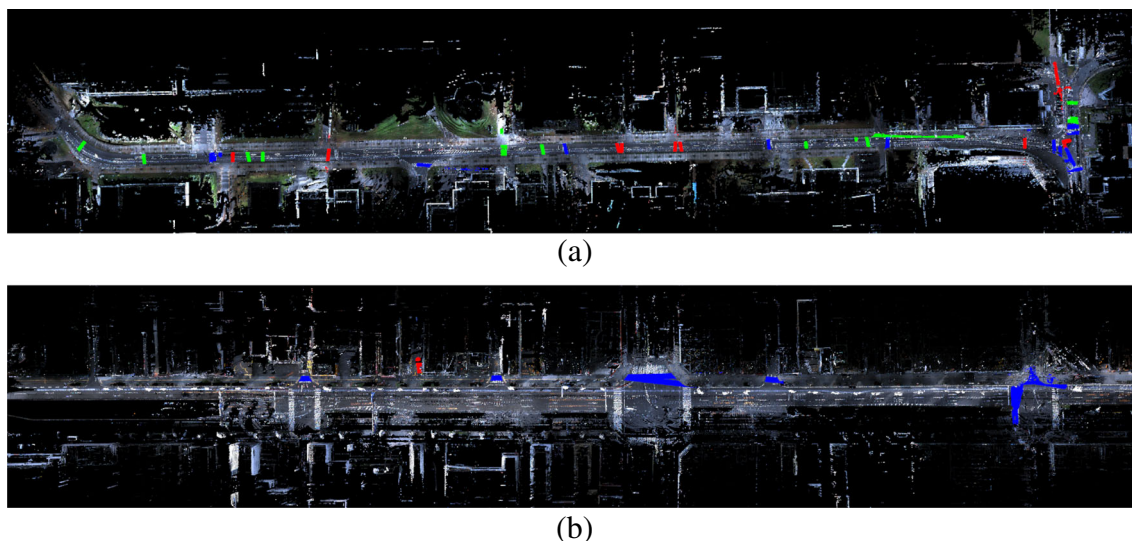


Fig. 24 RGB reconstruction of the environment of regions where pedestrians crossed the street detected by the algorithm. Where the green areas represent the regions observed by the authors, the blue

areas represent crosswalk regions, and the red areas represent new regions detected by the algorithm: **a** Experiment A; **b** Experiment B

pedestrians crossing and pedestrians at the other side of the street. Similarly, one can also notice an increase in building cross percentage on experiment B due to a higher number of buildings surrounding the crosswalk.

Regarding the detection of the pedestrian crossing, for the experiment A, four regions were not detected by the proposed approach. This indicates the absence of people walking along these regions. Among these four regions, three were crosswalks, which may not have been suitable for pedestrians. Crosswalk spacing criteria would be better determined according to the pedestrian flow and their preferences to cross the street. In general, the more distant a crosswalk is, or the longer the detour to reach it, the more likely a pedestrian is to jaywalk. By analyzing the crossing patterns determined by the pedestrian flow, it is possible to decide upon reallocating a crosswalk or adding a crosswalk in a specific location. Moreover, in the 21 ground truth regions, our algorithm detected 8 additional regions that could be more suitable for the location of a crosswalk than the three crosswalks that went undetected by our algorithm. For experiment B, five crosswalk regions went undetected, corresponding to those crosswalks where pedestrian movement can only be detected when the probe vehicle stops at a traffic sign. Also, from experiment B, the algorithm detected one new region, illustrated in red in Fig. 23. It corresponded to a parking lot area marked as the road in the semantic map. Since experiment B corresponds to a high traffic flux city area, one can expect not to find new crossing regions on a three-lane road.

State of the art methods for estimating pedestrian intention employ pedestrian as well as environment information. The location where pedestrians tend to cross a

street can be used as new information from the environment to improve the precision and accuracy of determining whether or not a pedestrian crosses the street. Therefore, identifying locations where pedestrians are likely to cross a street can improve autonomous cars' safety. For example, the locations where pedestrians are likely to cross can be used to improve the estimation of pedestrian crossing intention systems and generate a safe path planning for autonomous vehicles minimizing the risk of accidents. Moreover, these locations can also be used to infer the optimal location of crosswalks in a region according to the pedestrian tendency to cross the street.

8 Conclusion

This paper presents a method to extract information from dynamic traffic participants using sparse data and a semantic map obtained via an electric vehicle. A graph structure was used to represent pedestrian flow, wherein each node represented a region of pedestrian observation, and each edge represented the flow between two clusters. Moreover, the link between two clusters was determined through a probabilistic approach, where the heading direction extracted from pedestrian data was used to generate a GMM of the pedestrian's direction of movement in each region. After analyzing this graph, the edges crossing the road were extracted by comparing the edges with semantic information of the road.

The results show that the proposed system is capable of extracting information regarding pedestrian crossing patterns. The pedestrians estimated crossing patterns were

able to detect 81% of the regions where pedestrians have been seen crossing the street. This indicates that the algorithm can successfully identify the tendency of pedestrians crossing the street in an environment. This tendency of pedestrian crossing can further be used to improve the detection of pedestrian crossing intention, which would improve autonomous vehicles' safety.

For future works, first, the authors plan on developing a crosswalk extraction method to use crosswalk information on graph refinement. There are several types of crosswalks; in broad intersections, some of those crosswalks allow pedestrians to cross the street diagonally. One of the assumptions in this algorithm was that pedestrians only cross the street perpendicularly to the road. Therefore our algorithm would only show perpendicular crossing patterns, even if such patterns are allowed by a crosswalk. By developing a crosswalk extraction method for graph refinement, we could obtain a better analysis, which would lead to a broader display of pedestrian patterns. Second, the authors plan to extract GPS data for data acquisition to develop a more precise multiple daily data fusion method. At the moment, using only LiDAR information to generate a map using SLAM renders the map susceptible to noise during the SLAM process. This builds it up to a large cumulative error over time, which would affect the data fusion.

Author Contributions All authors contributed to the study conception and design. Material preparation, data collection and analysis were performed by RPN, KO and TW. The first draft of the manuscript was written by RPN and all authors commented on previous versions of the manuscript. All authors read and approved the final manuscript.

Funding This work was partially supported by JST CREST Recognition, Summarization, and Retrieval of Large-Scale Multimedia Data Grant Number JP14532298, NEDO's commissioned research project, Integration at the core of next-generation artificial intelligence and robots / 1. Research and development, Demonstration for social implementation of artificial intelligence technology / Research and development of transportation of soil at local small and medium construction sites using robot technology and artificial intelligence", and Advanced Intelligence Project (AIP) Challenge Research Funding, Japan.

Availability of Data and Materials All data and materials as well as software application or custom code support this published claims comply with field standards. The data that support the findings of this study are available from the corresponding author, RPN, upon reasonable request.

Declarations

Conflict of Interests The authors declare that they have no conflict of interest.

Open Access This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this

article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by/4.0/>.

References

1. Rasouli, A., Tsotsos, J.K.: Autonomous vehicles that interact with pedestrians: a survey of theory and practice. *IEEE Trans. Intell. Transp. Syst.*, pp. 1–19 (2019)
2. Bezerra, R.P., Ohno, K., Westfechtel, T., Tadokoro, S.: Pedestrian Flow Estimation Using Sparse Observation for Autonomous Vehicles. In: 2019 19Th International Conference on Advanced Robotics (ICAR). pp. 779–784 (2019)
3. Yao, C.Z., Lin, J.N.: A study of human mobility behavior dynamics: a perspective of a single vehicle with taxi. *Transportation Research Part A: Policy and Practice* **87**, 51–58 (2016)
4. Chen, G., Viana, A.C., Fiore, M., Sarraute, C.: Complete trajectory reconstruction from sparse mobile phone data. *EPJ Data Science* **8**(1), 30 (2019)
5. Prentow, T., Thom, A., Blunck, H., Vahrenhold, J.: Making Sense of Trajectory Data in Indoor Spaces. In: 2015 16Th IEEE International Conference on Mobile Data Management. Vol. 1, pp. 116–121 (2015)
6. Jahromi, K.K., Zignani, M., Gaito, S., Rossi, G.P.: Simulating human mobility patterns in urban areas. *Simul. Model. Pract. Theory* **62**, 137–156 (2016)
7. Nabavi Larijani, A., Olteanu Raimond, A.M., Perret, J., Brédif, M., Ziemlicki, C.: Investigating the mobile phone data to estimate the origin destination flow and analysis; case study: Paris region (2015)
8. Qiao, Y., Cheng, Y., Yang, J., Liu, J., Kato, N.: A mobility analytical framework for big mobile data in densely populated area. *IEEE Trans. Veh. Technol.* **66**(2), 1443–1455 (2017)
9. Zhou, J., Murphy, E., Long, Y.: Commuting efficiency in the Beijing metropolitan area: an exploration combining smartcard and travel survey data. *J. Transp. Geogr.* **41**, 175–183 (2014)
10. Briand, A.S., Côme, E., Trépanier, M., Oukhellou, L.: Analyzing year-to-year changes in public transport passenger behaviour using smart card data. *Transportation Research Part C: Emerging Technologies* **79**, 274–289 (2017)
11. Ma, X., Liu, C., Wen, H., Wang, Y., Wu, Y.J.: Understanding commuting patterns using transit smart card data. *J. Transp. Geogr.* **58**, 135–145 (2017)
12. Srikanth, S., Ansari, J.A., Ram, R.K., Sharma, S., Murthy, J.K., Krishna, K.M.: Infer: Intermediate Representations for Future Prediction. In: 2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). pp. 942–949 (2019)
13. Rhinehart, N., Kitani, K., Vernaza, P.: R2p2: A Reparameterized Pushforward Policy for Diverse Precise Generative Path Forecasting (2018)
14. Lee, N., Choi, W., Vernaza, P., Choy, C.B., Torr, P.H.S., Chandraker, M.: Desire: Distant Future Prediction in Dynamic Scenes with Interacting Agents. In: 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). pp. 2165–2174 (2017)
15. Karasev, V., Ayyaci, A., Heisele, B., Soatto, S.: Intent-Aware Long-Term Prediction of Pedestrian Motion. In: 2016 IEEE International Conference on Robotics and Automation (ICRA). pp. 2543–2549 (2016)

16. Hashemi, M.: Automatic inference of road and pedestrian networks from Spatial-Temporal trajectories. *IEEE Trans. Intell. Transp. Syst.* **20**(12), 4604–4620 (2019)
17. Higuchi, A., Takayama, S.: Dynamic Evaluation of Field Characteristics by Pedestrian Flow. In: 2017 Asian Control Conference, ASCC 2017. Vol. 2018-Janua, pp. 2569–2574. Institute of Electrical and Electronics Engineers Inc (2018)
18. Zhang, Y., Gao, K., Zhang, Y., Su, R.: Traffic light scheduling for Pedestrian-Vehicle Mixed-Flow networks. *IEEE Trans. Intell. Transp. Syst.* **20**(4), 1468–1483 (2019)
19. Morales, Y., Akai, N., Murase, H.: Personal Mobility Vehicle Autonomous Navigation through Pedestrian Flow: a Data Driven Approach for Parameter Extraction. In: IEEE International Conference on Intelligent Robots and Systems. pp. 3438–3444. Institute of Electrical and Electronics Engineers Inc (2018)
20. Frederico Carvalho, J., Vejdemo-Johansson, M., Pokorny, F.T., Kragic, D.: Long-term Prediction of Motion Trajectories Using Path Homology Clusters. *IEEE International Conference on Intelligent Robots and Systems*, pp. 765–772 (2019)
21. Qin, B., Chong, Z.J., Bandyopadhyay, T., Ang, M.H., Frazzoli, E., Rus, D.: Learning pedestrian activities for semantic mapping. In: Proceedings - IEEE International Conference on Robotics and Automation. pp. 6062–6069. Institute of Electrical and Electronics Engineers Inc (2014)
22. Kooij, J.F.P., Schneider, N., Flohr, F., Gavrila, D.M.: Context-Based Pedestrian Path Prediction. In: Fleet, D., Pajdla, T., Schiele, B., Tuytelaars, T. (eds.) *Computer Vision – ECCV 2014*, pp. 618–633. Springer International Publishing, Cham (2014)
23. Schneemann, F., Heinemann, P.: Context-Based Detection of Pedestrian Crossing Intention for Autonomous Driving in Urban Environments. In: 2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). pp. 2243–2248 (2016)
24. Rangesh, A., Trivedi, M.: When vehicles see pedestrians with phones: a multi-cue framework for recognizing phone-based activities of pedestrians *IEEE Transactions on Intelligent Vehicles* PP (2018)
25. Fang, Z., Ló, p.ez., A.: Is the pedestrian going to cross? answering by 2d pose estimation, pp. 1271–1276 (2018)
26. Rasouli, A., Kotseruba, I., Tsotsos, J.K.: Are they Going to Cross? a Benchmark Dataset and Baseline for Pedestrian Crosswalk Behavior. In: 2017 IEEE International Conference on Computer Vision Workshops (ICCVW). pp. 206–213 (2017)
27. Bonnin, S., Weisswange, T.H., Kummert, F., Schmuëdderich, J.: Pedestrian Crossing Prediction Using Multiple Context-Based Models. In: 17Th International IEEE Conference on Intelligent Transportation Systems (ITSC). pp. 378–385 (2014)
28. Westfechtel, T., Ohno, K., Bezerra, R., Kojima, S., Tadokoro, S.: Fusion of Camera and Lidar Data for Large Scale Semantic Mapping *IEEE Intelligent Transportation Systems Conference* (forthcoming) (2019)
29. Rusu, R.B., Marton, Z.C., Blodow, N., Dolha, M., Beetz, M.: Towards 3d point cloud based object maps for household environments. *Robot. Auton. Syst.* **56**(11), 927–941 (2008)
30. Kampker, A., Sefati, M., Rachman, A.S.A., Kreisköther, K., Campoy, P.: Towards multi-object detection and tracking in urban scenario under uncertainties. In: Proceedings of the 4th International Conference on Vehicle Technology and Intelligent Transport Systems - Volume 1: VEHITS, pp. 156–167. INSTICC, SciTePress (2018)
31. Biber, P., Strasser, W.: The normal distributions transform: a new approach to laser scan matching. In: Proceedings 2003 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2003) (Cat. No.03CH37453). vol. 3, pp. 2743–2748 vol.3 (2003)
32. Zhang, J., Singh, S.: Loam: Lidar odometry and mapping in real-time robotics: Science and Systems Conference (2014)

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Ranulfo Plutarco Bezerra Neto is currently pursuing a Ph.D. degree at Tohoku University, Japan. He received his M.Sc. and B.Sc. in computer science from the Federal University of Piauí, Brazil in 2018, and 2016 respectively. His research interests are robot intelligence and robotic perception. A member of RSJ and IEEE.

Kazunori Ohno received BS, MS, and Ph.D. from Tsukuba University in 1999, 2001, and 2004. He was a COE researcher at Kobe University in 2004, became an assistant professor and a lecturer at Tohoku University in 2005, and has been an associate professor of New Industry Creation Hatchery Center (NICHe) Tohoku University since 2012. He was also a PRESTO researcher (2008–2012) and has been a visiting researcher of the RIKEN Center of AIP (2017–now). His research fields are field robotics, robot intelligence, and cyber-enhanced canines. He established TC on Data Engineering Robotics of RSJ in 2012. He received Kisoi awards in 2008 and 2012, RSJ research awards in 2005 and 2019. A member of RSJ, JSME, VRSJ, JSAE, IEEE.

Thomas Westfechtel received his Ph.D. degree in Information Sciences from Tohoku University in 2019. He received his M.Sc. And B.Sc. in electrical engineering from Paderborn University in 2015, and 2012 respectively. He is currently a special appointed assistant professor at the Research Center for Advanced Science and Technology (RCAST) of the University of Tokyo. His research interests are machine intelligence, robotic perception and knowledge acquisition for autonomous ground vehicles.

Shotaro Kojima received BS, MS, and Ph.D. from Tohoku University in 2015, 2017, and 2021. He was a DC2 research fellow of Japan Society for the Promotion of Science (2018–2019). He was a research fellow of New Industry Creation Hatchery Center, Tohoku University (2020). He is an assistant professor of New Industry Creation Hatchery Center, Tohoku University (2021–). A member of RSJ, JSME, and IEEE.

Kento Yamada received BS, MS. from Tohoku University in 2019, 2021. His research interest is autonomous construction vehicles.

Satoshi Tadokoro graduated from the University of Tokyo in 1984. He was an associate professor in Kobe University in 1993–2005, and has been a Professor of Tohoku University since 2005. He was a Vice/Deputy Dean of Graduate School of Information Sciences in 2012–14, and is the Director of Tough Cyberphysical AI Research Center since 2019 in Tohoku University. He has been the President of International Rescue System Institute since 2002, and was the President of IEEE Robotics and Automation Society in 2016–17. He served as a program manager of MEXT DDT Project on rescue robotics in 2002–07, and was a project manager of Japan Cabinet Office ImPACT Tough Robotics Challenge Project on disaster robotics in 2014–19 having 62 international PIs and 300 researchers that created Cyber Rescue Canine, Dragon Firefighter, etc. His research team in Tohoku University has developed various rescue robots, two of which called Quince and Active Scope Camera are widely-recognized for their contribution to disaster response including missions in the Fukushima-Daiichi NPP nuclear reactor buildings. IEEE Fellow, RSJ Fellow, JSME Fellow, and SICE Fellow.