

Person Tracking Using Ankle-Level LiDAR Based on Enhanced DBSCAN and OPTICS

Mahmudul Hasan^{*,**a}, Non-member

Junichi Hanawa^{*}, Non-member

Riku Goto^{*}, Non-member

Hisato Fukuda^{*}, Non-member

Yoshinori Kuno^{*}, Member

Yoshinori Kobayashi^{*}, Non-Member

Along with the progress of deep learning techniques, people tracking using video cameras became easy and accurate. However, privacy and security issues are not enough to be concerned with vision-based monitoring. People may not be tolerated surveillance cameras installed everywhere in our daily life. A camera-based system may not work robustly in unusual situations such as smoke, fogs, or darkness. To cope with these problems, we propose a two-dimensional (2D) LiDAR-based people tracking technique based on clustering algorithms. A LiDAR sensor is a prominent approach for tracking people without disclosing their identity, even under challenging conditions. For tracking people, we propose modified density-based spatial clustering of applications with noise (DBSCAN) and ordering points to identify cluster structure (OPTICS) algorithms for clustering 2D LiDAR data. We have confirmed that our approach significantly improves the accuracy and robustness of people tracking through the experiments. © 2021 Institute of Electrical Engineers of Japan. Published by Wiley Periodicals LLC.

Keywords: person tracking; DBSCAN; OPTICS; density-based clustering; LiDAR

Received 30 October 2020; Revised 1 February 2021

1. Introduction

Nowadays, people are under surveillance in their office, on roads even at home. To realize a safe and secure society protecting civilians from different offenses, we must keep everything under supervision. On the other hand, peoples are claiming their privacy and secrecy because surveillance videos are possibly leaked. Services in public places like airports, stations, museums, and super-shops require people trajectories to provide higher quality services. Our two-dimensional (2D) LiDAR-based tracking system can track individuals robustly and accurately. When individual identification is not a concern but only tracking a person is necessary, our system plays a vital role. A wide range of views and low computational costs are also the benefits of 2D LiDAR-based tracking. Using video cameras, three-dimensional (3D) LiDAR, and deep neural networks (DNN) makes the system pricey in cost and time. This application is highly applicable for emergencies, especially for rescue systems in natural disasters. The 2D LiDAR sensor may be more worthy over RGB/RGB-D cameras in terms of the fast computation and robustness against occlusion.

Video-based person tracking is a wide-spread technology and is accepted broadly in different applications. We replaced cameras with 2D LiDAR and tried to find an alternative. We put our 2D LiDAR at the lower leg (ankle) level and gathered distance information in 270° headings. We plot the distance data on the 2D image plane like a top view (we call this image a “LiDAR image”). Here, using the background subtraction technique, the system can eliminate structural information such as walls or stable objects. Finally, we get only the position data of moving objects drawn on the LiDAR image. For tracking individuals, it is necessary to discriminate data points belonging to one person somehow. Generally, the method takes two steps: the data points are clustered for each ankle, and then two ankles are paired for one person. In other words, we measured the distance between the pixels to create clusters to decide an ankle. The distance between two ankles is then calculated to discriminate individuals. In this approach, there are several difficulties. Firstly, two-step clustering is a time-consuming process. Processing speed is a significant factor for tracking; Secondly, after finding multiple persons’ ankles, it is challenging to recognize which two ankles belong to one person in the multiple person context. In constant, our method proposed in this article can identify individuals using only one step clustering. Usually, two ankles belonging to one person are passing each other closely while walking. Therefore, we make motion history images from 2D LiDAR data (time series of distance data plotted on one image), including the frame of closely passing two ankles. Then we apply one-step clustering to discriminate individuals.

^a Correspondence to: Mahmudul Hasan. E-mail: hasan@hci.ics.saitama-u.ac.jp

^{*}HCI Lab, Saitama University, Saitama, Japan

^{**}Department of CSE, Comilla University, Cumilla, 3506, Bangladesh

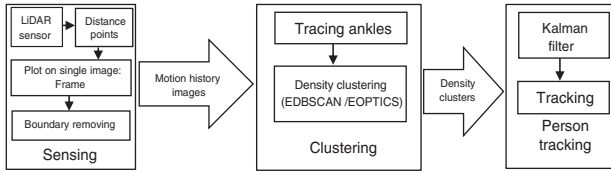


Fig. 1. Block diagram of LiDAR-based person tracking

For this data clustering, we previously used traditional clustering algorithms. Here, we have used density-based algorithms. Density-based clustering has a tremendous influence on finding the clusters with random contours and easy noise cancellation techniques. Another significant advantage of the density-based algorithm is not to set several clusters previously. It makes clusters based on data density. We modified two well-known density-based algorithms density-based spatial clustering of applications with noise (DBSCAN) and ordering points to identify cluster structure (OPTICS), for our system by considering the maxPts parameter. This modification makes these algorithms more robust for person tracking. We interestingly see that these enhanced algorithms time complexities are as previous, and performance increased with a considerable amount of time.

In Fig. 1, we depicted the block diagram of our LiDAR-based person tracking. Three individual sections work together to track a person. The first section is sensing. A 2D LiDAR sensor is used for getting distance data. These distance data of a particular time are plotting on an image (we call this a “frame”). We applied a boundary removing technique for removing unwanted data from the sensing image. Ankles are traced from motion history images. After getting ankle positions in the frame, we applied our modified algorithm for clustering to accurately track a person. Here, the Kalman filter (KF) is used for tracking the person. Our primary focus in this study is to find a sophisticated person tracking system using 2D LiDAR data. To increase the system accuracy, we included two density-based algorithms separately and showed that enhanced algorithms provide more accurate tracking results. Either EDBSCAN or EOPTICS can be used for our clustering purpose. Both algorithms have good effectiveness compared with others, although EOPTICS performs better here. This study will also help to find clustering algorithms in time series 2D LiDAR data processing.

2. Related Works

2.1. Related works in sensor-based tracking Many people worked on person tracking based on different sensor-based techniques. Video-based person tracking [1] is a common phenomenon in this regard. Improvement of technology and processing techniques make this research accurate day by day. The 3D skeleton-based person tracking [2] gave a novel idea in this research. Here they described a 3D skeletal model for recognition and tracking based on a topology-based framework. Surveillance cameras are needed for video recording. Some researchers used Recurrent Convolutional Network (RCN) for person identification [3]. Here recurrent neural network architecture was used for video analysis and re-identification. Multiple people tracking from videos are necessary for continuous surveillance [4]. This tracking should be instant and accurate. Rather than video, some researchers tried to develop sensor-based person tracking. Radio-based multiple sensors [5] was used for tracking. In this research,

they combined the guard system and sensing of body motion for person tracking. Usually, these transformations take unnecessary time for positioning. Furthermore, using multiple sensors enhances the complexity of the system.

In multiple people tracking systems, occlusions are one of the fundamental problems. In a video-based approach, sometimes employing shadows were used for multiperson tracking [6]. Using multiple cameras and their person tracking, and indexing is unnecessarily expensive and limited to computational ability. An unsupervised feature selection mechanism based on multi-view architecture [7] was proposed in this circumstance. Here they presented a distributed coding for feature selection and exhibited its use in multiviewed recognition. New sensors were developed over the period, and scientists tried to use these in different applications. A person tracking framework using a low-resolution infrared thermal sensor [8] is one of these. Here they combined RGB and infrared cameras for tracking. We focused only on one sensor-based tracking for our research and tried to find a replacement for cameras. But this approach is based on multisensor applications. Over the period, researchers tried to find out some alternatives to cameras. LiDAR-based gait analysis [9] and different activities of humans for surveillance was auspicious work. Their suggested algorithms are implanted in a visualization system that can analyze and collaboratively exhibit practical scenarios in outside situations where pedestrians are walking. They tried to investigate a gait-based person identification along with some activity recognition based on the LiDAR sensor.

2.2. Related works in cluster-based tracking For our research, we concentrated on density-based clustering. Here DBSCAN [10], OPTICS, and point-cloud [11] show significant improvements over others. However, little but some research was done with these algorithms on LiDAR sensor data. Eps estimation with DBSCAN for LIDAR data [12] was influential research. Here they concentrated on automatic point cloud segmentation and parameter estimation for DBSCAN. They evaluated their approach with point cloud data. Their method was not applied to pedestrian tracking and real-time LiDAR data. We focused on it and tried to investigate its impact on tracking. Laser scanning was used in different applications for a couple of decades. Some researchers performed contextual classification [13,14] of the LiDAR point cloud. Here they present a probabilistic method for contextual classification of point cloud data. Further, they tried to detect building objects based on the classified point cloud. Here discriminating features are not identified that caused some misclassifications. We tried to find optimistic results, so there. Some findings showed that HDBSCAN [15] is a more robust and modified density-based algorithm than DBSCAN. Here there is no need to consider one parameter of DBSCAN is the radius. HDBSCAN can set this as infinite, for instance. But in our modified EDBSCAN works differently on LiDAR-based datasets to calculate accurate tracking. HDBSCAN is not well suited for Euclidean space. Both HDBSCAN and DBSCAN are not fully deterministic. They are suffering from the curse of the dimensionality problem. Here are the proposed EDBSCAN and EOPTICS that play a substantial role.

In our previous work [16], we tried to establish a LiDAR-based tracking system that can track pedestrians properly. We used the DBSCAN algorithm for clustering the data and KF to track a person’s path. Different parameters of DBSCAN [17] were set according to experimental results. In this research, we tried

to enhance the accuracy of the tracking. We changed traditional clustering DBSCAN with an enhancement (*E*-DBSCAN) and incorporated enhanced OPTICS for optimal performance. We found significant development of our previous experiments using these two improved clustering approaches.

3. Methodology

To track a person is a very critical decision about where to place the sensor. Hence, establishing a LiDAR sensor in an ankle position can be the right solution. Because LiDAR can track every ankle, which will help to determine walking paths. Here it can avoid people's body occlusion, and it has other applications also. The mobile robot usually puts LiDAR at ankle level because of obstacle detection.

It is challenging to track a person independently. When multiple peoples are walking on a surface, close ankle positions may misclassify. Figure 2(a) shows an apparent description of 2D LiDAR sensing. It provides horizontal distances of moving ankles in X and Y coordinates relative to its position. These distance values are plot on a frame, and we can get deals as Fig. 2(b). An individual clustering approach is performed of these distance values to take a decision of an ankle and chronologically two pairwise ankles make a person. From Fig. 2(b), we see that some peoples are walking on a plane. In Fig. 2(c), red-marked circles indicate a person where two close ankles are acted like a person. The fourth Fig. 2(d) calculates the cluster differently. This is an anomaly of clustering and occurs misclassification.

Sensing data from a LiDAR and its better classifications were our primary objective. We repeatedly placed the sensor in different positions from 10 to 200 cm. Usually, human leg lengths vary from 0 to 100 cm. To get these data and significant influences, we placed our LiDAR sensor at 20 cm height. We see that this height provides the most accurate data for tracking. With fantastic accuracy over single pedestrian movements, we faced a downfall in a group movement. Which ankle belongs to which circle, taking this decision was troublesome indeed. We repeatedly changed cluster thresholds of traditional clustering approaches, but problems were persisted. For better approximation and accuracy, we introduced a state of art tracking method based on density-based clustering. Here we able to solve wrong clustering scenarios.

3.1. Sensing We used the HOKUYO UMT30LX LiDAR sensor for data acquisition. For our experiments, we keep our sensor static and considered objects are moving. Here objects refer to the pedestrian. In the below picture (Fig. 3), we held our sensor in an ankle position and captured time-series video data using LiDAR. These data were being stored for further surveillance. We applied some preprocessing techniques to make the data more robust and accurate. In the second part of the picture, we see that a LiDAR-based image is placed with a white marked surface boundary. Here centered white marked dots show the ankle positions of a person in a specific frame. We tried to remove the white background from captured images. It helped us to handle only ankle positions on the frame without unwanted noise. We applied the background subtraction method for removing boundaries. After that, we got only the ankles marked frames of every individual moment. Sequences of those frames created a LiDAR-based video.

Feature extraction from preprocessed data is another big concern in this research. We considered sequential frames for feature

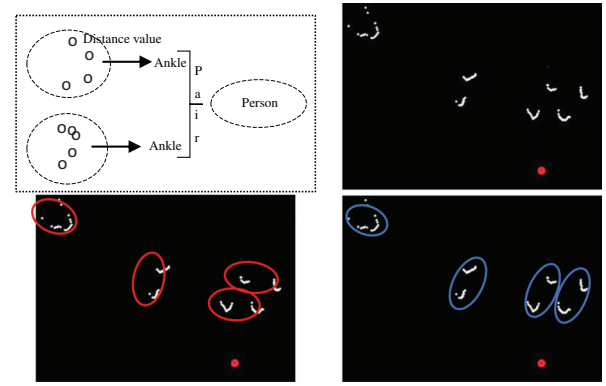


Fig. 2. (a) Different distance values of ankles; (b) LiDAR-based ankle positions on the frame; (c) considering close ankles as a person; and (d) ankles in different orientations make a cluster

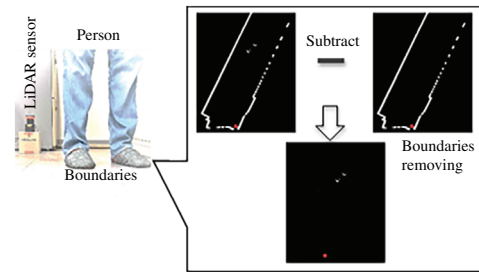


Fig. 3. Image sensing using LiDAR sensor and preprocessing

selection. Every chronological data represents every ankle position on the frame. Movements of an ankle and two ankles of a person were considered as a person. But critical situations always appeared in data sensing. Occlusion, very fast-moving ankles, and omnidirectional movements of a pedestrian all these challenges make this study more complicated and troublesome. To cope up with these shortcomings and make the system more capable concerning time and calculation, we thought about using a second LiDAR sensor in another direction on the plane. Using another sensor makes the system inefficient and causes a higher installation cost. We changed our experimental architecture and decided to focus on a single sensor-based diagram. Here we emphasized making a vigorous development of algorithms that can handle all these deficiencies.

3.2. Clustering The next section of this research is to cluster the data so that it can handle all critical scenarios. We consider density-based algorithms. Density-based clustering can find random-shape clusters also can manage noises. In density-based clustering approaches, clusters are created according to dense areas that are distinct from sparse areas. These algorithms work on data density. Here no previous assumption is needed. Clusters are made based on data density. DBSCAN is one of the popular density-based clustering algorithms we used in our research. We also applied the OPTICS algorithm in this research. Both algorithms performed very well on person tracking. With many practical benefits of these two algorithms, still, some constraints must be fixed. We tried to solve these issues and proposed a modification of these two algorithms. Enhanced-DBSCAN and Enhanced-OPTICS

algorithms are the transformation of traditional algorithms but can perform better on LiDAR-based tracking.

3.2.1. Enhanced (E) DBSCAN Density-based spatial clustering of applications with noise (DBSCAN) is devised to locate nonsphere-shaped clusters. Two specific parameters: first, radius ϵ , a precise distance that defines the neighborhood, and the second one is the minimum number of points (minPts) that are considered as neighbor points, determine the dense neighborhood of DBSCAN. DBSCAN is an exciting development to make a cluster based on LiDAR data. But to make a robust tracking system, DBSCAN is still facing some limitations that degrade tracking performance. Selecting initial parameters (ϵ , minPts) is one drawback of this algorithm. But it has a high computational cost of $O(n^2)$, which affects real-time computations. Compared with other shortcomings of ordinary hierarchical algorithms, we find DBSCAN as a better performing algorithm, especially in LiDAR data processing. Here to track an ankle, we successfully set its initial parameters that work without disappointments in most cases. Furthermore, we used a LiDAR sensor and applied background subtraction that reduced data congestion and processing. So, our system was rated and functional with real-time pedestrian tracking.

We faced another difficulty in using DBSCAN. When multiple peoples are walking on the surface, their ankles come nearer. As we described, DBSCAN creates clusters based on minPts and ϵ , ankles of different persons come so closer that the difference between two ankles of other people's seems like a single person's ankle, where the difference value is lower than the radius. Here, DBSCAN fails. It creates a cluster with all closer ankles within its range, even if it belongs to different persons. Changing the thresholds of the radius may solve the problem. But in a practical application, the radius cannot be altered based on crowd size and movements. So, we must refrain from the structure of DBSCAN that it best fits with our application and solve this drawback of data overfitting.

From the images (Fig. 4), we can get a clear idea about the shortcomings of DBSCAN. The left topmost image shows different ankle positions on the frame. The corresponding lowermost image creates a cluster based on DBSCAN parameter minPts and radius ϵ . Here we set our parameters as minPts = 20 and ϵ = 900, which means DBSCAN will create a cluster if it gets at least 20 points within the radius ϵ and determine as a person. From the first image, we see four persons walking on the plane, but the corresponding lowermost image detects it 5. Here fifth person's two ankles distance is more significant than our threshold ϵ , but every ankle satisfies the condition of being a cluster that makes two clusters and shows as two persons. After two frames later, in the second figure of the top row, it recalculates the distances, creates four clusters, and shows the corresponding lowermost image. In the third frame of the top row, persons come closer two frames later. Here their ankles are occluded by one another. In this situation, the ankle distance between two or more persons comes lower than ϵ , which creates a single cluster, an example of data overfitting. Here in the second row, the last image shows are the result of two persons. This is a drawback of the straight use of DBSCAN.

To solve these issues, we set the third parameter for the DBSCAN algorithm. We consider maximum points (maxPts) here and two previous parameters (minPts and ϵ). Setting this parameter can quickly reduce data overfitting and fixed radius problems. Suppose 10 points can create an ankle, then at least 20 points

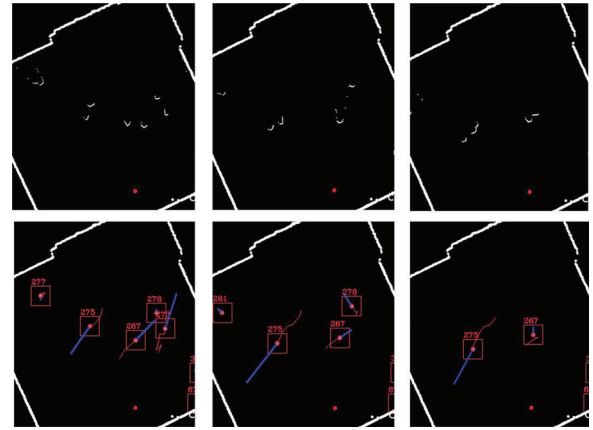


Fig. 4. The uppermost images show an ankle position on the frame—Lowermost images pointing persons with the handler

will be needed to create a cluster to be a person. Then minPts of DBSCAN can be 20. Again, we can set the distance ϵ between two points using Euclidean distance as 900 units. If four ankles of two people come closer and their points distance lies within 900 units, then previous DBSCAN considers these as a single cluster. DBSCAN thinks about minimum data by which it can create a cluster. Here there is no obligation of maximum data. But LiDAR based tracking fails in this concept. If we set maxPts, another parameter, like 50, this DBSCAN can create a cluster of data points lying within 20–50. As less than 20 points will not make a cluster parallelly, if it goes a maximum of 50, it will also discard the cluster. Adopting this parameter can solve the data overfitting problem. If more than two ankles come closer and their combined number of points exceeds maxPts, Enhanced DBSCAN (EDBSCAN) will not accept this cluster. EDBSCAN discards this cluster immediately. But there is still another problem that exists: to create clusters again with its overfitted data. Here EDBSCAN changes its radius ϵ immediately. It decreases its radius by a significant amount and finds clusters with discarded points, not in the whole dataset. This step will continue until the overfitting problem is solved. Here minPts and maxPts values will remain the same; the only radius is decreased.

In Fig. 5, within the radius ϵ , there are so many points within the circle. EDBSCAN will discard this cluster as it has so many data points than maxPts. This method justifies the concept that more than one person is walking in front of the LiDAR sensor whose ankles come closer. Here the bottleneck problem of traditional DBSCAN can be solved by our proposed Enhanced DBSCAN. Now another proposed portion of EDBSCAN is varied radius selection. As the initial selection of radius creates a data overfitting problem, we can reduce the radius with a significant amount where minPts and maxPts parameters will be the same as previous. A person can be identified again in a crowded situation. In the figure below (Fig. 6), radius ϵ is reduced to ϵ' , now it can find a new cluster with initial minPts 20 and maxPts 50, which will see those clusters that can satisfy the condition. Here although, the ankles of different persons are so close that there is a chance of misclassification, but a reduced radius can quickly solve the issue. Only very close points will make a cluster.

Setting max point (maxPts) parameters in DBSCAN, we can also handle false clustering problems. Suppose there is a moving object on the plane rather than an ankle. If it creates with required

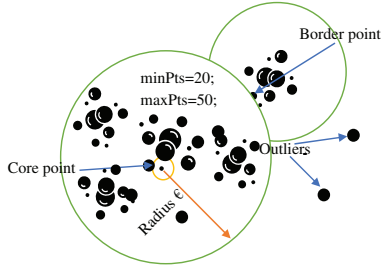


Fig. 5. Enhanced DBSCAN (EDBSCAN) mechanism

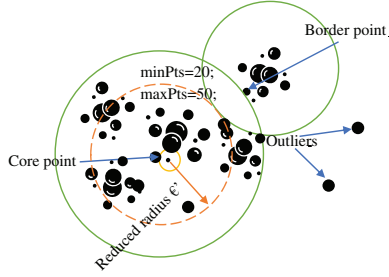


Fig. 6. Reduced radius of EDBSCAN

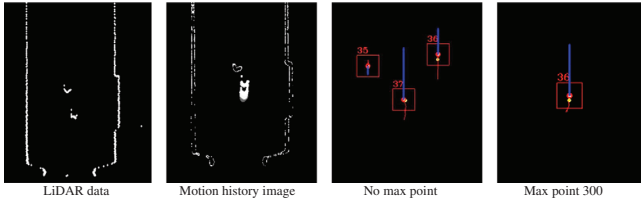


Fig. 7. Anomaly in tracking without using maxPts

minPts and ϵ , traditional DBSCAN will create a cluster as a person moves on the plane. Our proposed maxPts parameter can handle it easily. It produces clusters based on motion history images. It will discard all false clusters if it goes more than maxPts and does not appear on consecutive images.

Suppose in the Fig. 7; there are four frames of a moment of a person walking in front of the LiDAR sensor. In the first frame, this is LiDAR data captured by the sensor. The second frame shows a motion history image of that moment. When conventional DBSCAN was used to track a person, there was an anomaly in tracking shows in the third frame. From the previous frames, we see that only one person is walking on the frame. But due to garbage data and false moving of objects, DBSCAN detects another two persons on the frame. When we used the maxPts parameter with value 300, our EDBSCAN removed erroneous classification and showed only one person in the frame as original data.

3.2.2. Enhanced (E) OPTICS OPTICS is very close to DBSCAN but refers to another procedure. This is a new approach based on data density to make a cluster on spatial data. When data density varies, DBSCAN has a weak performance on data for making clustering. OPTICS addresses this drawback and can create profound clusters with differing densities. It works in a way like the previous one in that all points in the input data are ordered as core points, border points, and outliers. OPTICS also works with

two input parameters: maximum distance to be considered, radius ϵ , and a minimum number of points to create a cluster, minPts. The reachability distance R_d between core points Q and S can be measured as:

$$R_d = \max \{ \text{Dist}(Q, S), \text{Core Dist}(Q) \}$$

OPTICS works on the same cluster, which can find subclusters within the cluster. This approach can find out the inner trends of clusters within the central cluster. We have proposed the development of OPTICS as Enhanced OPTICS named EOPTICS, where another parameter is applied as maximum data points named maxPts along with previous minPts and Radius ϵ . The idea is exciting and aligned with our practical application, LiDAR-based tracking. To make a cluster data point should be within the minPts, maxPts, and radius ϵ . Our approach is to create a density-based cluster that does not exceed maxPts with its data point. The remaining process will be the same as OPTICS. Here it will check cluster density. This approach is a clear development and significant modification of OPTICS. Here highly dense areas where more than one people's ankles come closer and traditional OPTICS cluster creates a single cluster with all data points that can be ignored. Now enhancement of OPTICS has another significant benefit. If several points exceed maxPts, then our algorithm discards the outer cluster that means the main cluster. The inner clusters based on core distances and reachability distance will be calculated here. These internal clusters will help to determine the ankle's positions on the plane. Our EOPTICS then reduces radius ϵ with a significant amount and compute clusters again. Inner clusters will also be calculated to find the associativity of data within the main cluster. This approach will help to find correlated subclusters that imply a person on the plane.

In all cases, cluster parameters will be calculated based on some critical criteria:

- If the cluster radius creates only one cluster, it will calculate core distance and find subclusters. If there is more than one subcluster, it will discard the outer main subcluster and consider inner subclusters as the main cluster. If only a few clusters were created with very few minPts, it will increase its radius ϵ and recalculate clusters again.
- If for n minPts EOPTICS creates m cluster where $m \geq \frac{D}{2}$ then discard the cluster and recalculate minPts as $\text{minPts} = 2 * \text{minPts} < \text{maxPts}$. The process will continue until $m < \frac{D}{2}$, where, D is data points.
- If there is no cluster found with the radius ϵ , EOPTICS increases ϵ and tries to calculate the cluster. If more than one cluster is found, then it will keep it; discard otherwise.

In Fig. 8, we can see the traditional approaches of the EOPTICS algorithm. This is close to OPTICS. It gets ankle positions from the sensor and makes these clusters. Here in the left picture, we see a certain frame where four peoples are moving in front of the LiDAR sensor. KF-based tracking tracks all these persons based on data. Parameter maxPts setting is a heuristic process. For our experiments, we varied this from 50 to 300 points. It depends on crowd size and frequency on the surface.

A moving direction is also shown in which direction he/she is moving. After one frame later, one ankle of a person goes out of range of the sensor, and two people come closer than their ankles distance lies within the scope of our radius. Ordinary OPTICS

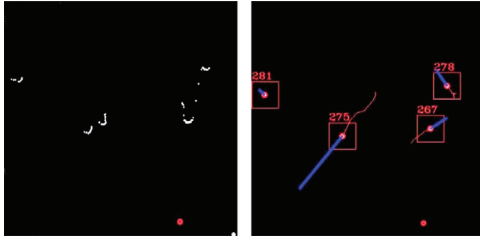


Fig. 8. EOPTICS with LiDAR data; (a) ankle positions in front of the sensor, (b) KF-based tracking

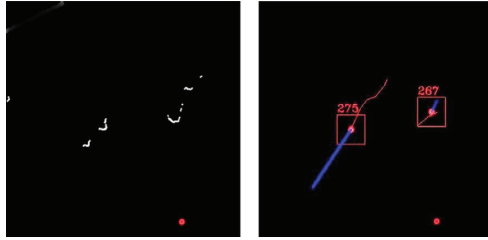


Fig. 9. OPTICS; (a) one frame later, ankle positions, (b) corresponding Kalman-based tracking

creates only two clusters here. One cluster is omitted because of a shortage of minPts parameter. Here only one ankle cannot fulfill minPts requirements. The second cluster was created with no anomalies that show ID 275 in Fig. 9(b). The critical condition appeared in the next section. Here two persons come closer. In the previous frame, they have identified accurately with ID 267 and 278. But ordinary OPTICS created here only one cluster 267 and showed it as a single person. This is the shortcoming of OPTICS.

Our proposed EOPTICS plays a very significant role here with this inadequacy. In the figure below (Fig. 10), we see that there is a very substantial enhancement over OPTICS; EOPTICS gave three clusters where the KF provides three persons on the frame. For the same reason, OPTICS rejected one cluster here; EOPTICS rejected it for the same that the data points are less than one parameter minPts. The second cluster was created as previously.

Here the data points lie within our range greater than and less than minPts and maxPts, respectively. EOPTICS revised the third cluster than OPTICS. When OPTICS created only a single cluster with two person's ankle points, EOPTICS denied making this a single cluster. Here the idea is, it counts minPts as well as maxPts. In this situation, data points satisfy the minPts condition but have more data points than maxPts. As condition dissatisfies, EOPTICS stopped creating a cluster. It works with its next phase. Now it reduced its radius ϵ and tried to develop clusters again only with these sorted data. After reducing ϵ , only relevant ankle points satisfy all the conditions and created two separate clusters. In the right-side figure, handler 267 and 278 showing these.

3.3. Tracking The next step of our research is tracking. The KF tracks the gravity point of the cluster. A forecast equation and a revised equation are the base of the KF.

$$\text{Forecast, } F(n) = XF(n-1) + S(n-1)$$

$$\text{Revise, } R(n) = YF(n) + M(n)$$

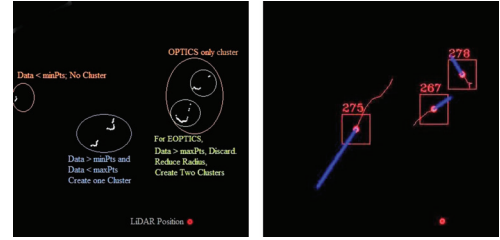


Fig. 10. EOPTICS, (a) ankle positions, (b) KF-based tracking

where, $F(n)$ and $R(n)$ are forecasting estimation and revise measurement variables, X and Y are State-Transition-Matrix and Measurement matrices, respectively. S represents system noise, and M represents measurement noise. Input motion history images are taken as input, and KF tries to forecast it. If it can detect, then forward it to the revise section. If any revision is needed, it again feedbacks it to the forecast section. This process will continue until the desired tracking is found. Finally, a person is being tracked.

4. Result and Discussion

As there were no well-established datasets on LiDAR-based person tracking, we prepared our datasets for experiments. LiDAR-based tracking always suffers from data misfit and low accuracy problems. We used a LiDAR sensor that suffers a lot in this situation. Initially, we use recorded video for our experiments. We found significant results there. Further, we enhanced our system for real-time tracking. Interestingly we get comparable improvements with the video-based person tracking system.

We performed our experiments with different male and female participants. We considered another group of peoples for our experiments. We choose five substantial categories for this analysis: Individual Walking, Individual Running, Ankle Movements Only, Combined Walking, and Combined Running. We applied our developed EDBSCAN to individuals and groups of peoples tracking. Here in Fig. 11, combined walking and running are focused. In the first frame, we see that three people are walking in front of the LiDAR sensor. Our progressive approach tracks these people and shows them in the rightmost picture. In the second picture, we see that four people are running, and our developed approach identifies all these people. Here EDBSCAN plays a very significant development over traditional DBSCAN. When more ankles come closer, it can prevent these from misclassification. maxPts parameter discards these clusters and recalculates again.

We have examined EOPTICS also for our experiments. The development of OPTICS as EOPTICS shows significant results in our system. We applied this algorithm on our same data sets and found EOPTICS most effective in the frame-by-frame analysis. In Fig. 12, we placed three gestures together and found EOPTICS functional than previous DBSCAN and OPTICS. Here left most pictures to show that ankle positions in the frame, and corresponding rightmost pictures show their movements and tracking based on KF. If any misclassification occurs in an individual frame, EOPTICS solves this in the next frames with its parameter update techniques. This same approach also applied to the combined walking and running of different groups of peoples.

We solved occlusion problems also. It occurs when one ankle comes in front of another, and LiDAR cannot get a signal from the next one due to obstacles. In a person tracking system, this occlusion problem creates a problem, and sometimes it is

Different gestures (EDBSCAN)	Ankle positions on the frame	Direction of the movements
Combined walking		
Combined running		

Fig. 11. EDBSCAN based group peoples tracking with LiDAR sensor

Different gestures (EOPTICS)	Ankle positions on the frame	Direction of the movements
Individual walking		
Individual running		
Only ankle movement		

Fig. 12. EOPTICS based individual tracking with LiDAR sensor

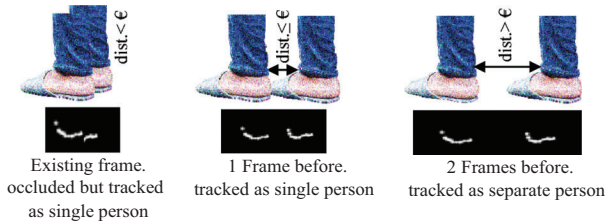


Fig. 13. Tracking and occlusion handling, a frame by frame analysis

not possible to track a person accurately for this scenario. Our proposed LiDAR-based tracking system can handle this problem. In Fig. 13, suppose two frames ago, our proposed algorithms calculated the distance between ankles and found this more than the acceptance threshold. This will decide it as two persons. In one frame back, both frames come closer, and their distance becomes less than our radius, and here it is updated as one person. This is the continuous up-gradation of tracking. If there are any anomalies occurred in any frame, but it can be modified in consecutive frames. These tracking anomalies can be accepted in an individual frame or very few frames for a person tracking.

Similarly, in the current frame, one ankle is occluded by another, and their ankle distance cannot be calculated. It happened because of a lower gap between ankles or occlusion. Here if the radius is less than the threshold, then there is no problem. It satisfies the condition. If data points are less than the maxPts parameter, then it will create a cluster and identify it as a person. If multiple people come so close that their ankle distance cannot be measured and the number of points goes over the maxPts, it will discard the cluster and recalculate the radius and form a cluster again.

In Table I. we have shown some experimental results of different gestures. Here five motions were considered. Another group of people participated in our experiments over a couple of minutes. We took random frames here for validation and verification. Suppose in the table below for individual walking and running three and four peoples have participated. We took 44 and 49 consecutive frames randomly. Based on our developed EDBSCAN and EOPTICS algorithms, we got almost correct recognition there.

Similarly, for only ankle movements, we got considerably better results here with 52 and 53 frames accurate recognitions out of 56. For combined walking and running, we took nine peoples in our experiments. In the case of the group walking here, EOPTICS performs better than EDBSCAN. But compared with previously developed algorithms, it is also impressive development. We have a good scope to improve our system for a combined running situation. If a group of people is running fast to and fro, it is not easy to trace with utmost accuracy. We considered 136 frames for our experiment where EOPTICS recognized 106 frames and EDBSCAN could 101 frames accurately with a total elapsed time of the algorithms are 32.41 and 27.63 s, respectively. Our used LiDAR sensor (UTM-30LX) took 25 ms per scan.

We compared our improved algorithms with traditional DBSCAN and OPTICS. In Table II, we have shown these performances and accuracy for better understanding. For individual walking, all four conventional and improved algorithms performed well. This scenario is the same for individuals running also. For only ankle movements, EOPTICS plays sizable improvements here. It can handle some misclassifications than others. These developments also persist in group walking and running. Almost in all cases, EOPTICS plays better results. EDBSCAN also performs well compared with previously applied density-based algorithms.

It is adorably admissible that typical hardcore clustering algorithms are not well suited with LiDAR-based person tracking. We previously performed and have shown here some experiments regarding this issue. Both EDBSCAN and EOPTICS apply to these types of data sets. Furthermore, improvements in DBSCAN and OPTICS do not heighten algorithm complexity. The worst-case complexity persists $O(n^2)$, and without matrix completion, it requires $O(n)$ memory only.

We compared our approaches with present advanced precisions (Table III). Different people used different clustering approaches and datasets for person tracking, but we wanted to show our improved EDBSCAN and EOPTICS precisions here. In 2011, Rusu et al. [11] stated a strategy for tracking with Euclidean Clustering. They got 64.5% accuracy with a cluster threshold of 0.5 m. In 2016, Depth Clustering [18] approach was applied for the same and found incredible 39.2% precision with a 10° entry. Another technique Run Clustering [19], was considered in 2017 and found 51.7% accuracy in 3D tracking. Online Learning [20] proposed an adaptive threshold selection method in 2020 and found 89.8% accuracy. Our previous approach used only DBSCAN

Table I. Experimented results for EDBSCAN and EOPTICS

Gestures	Persons	Frames	Correctly Identified by EDBSCAN	Correctly Identified by EOPTICS
Individual walking	3	44	44	44
Individual running	4	49	48	48
Ankle movement	5	56	52	53
Combined walking	9	131	123	127
Combined running	9	136	101	106

Table II. Comparison of EDBSCAN and EOPTICS with DBSCAN and OPTICS

Gestures	Accuracy with DBSCAN	Accuracy with OPTICS	Accuracy with EDBSCAN	Accuracy with EOPTICS
Individual walking	99%	99.5%	100%	100%
Individual running	98.01%	97.86%	97.96%	97.96%
Ankle movement	92.07%	92.02%	92.86%	94.64%
Combined walking	93.71%	93.81%	93.89%	96.95%
Combined running	71.87%	72.11%	74.26%	77.94%

Table III. Comparison with existing approaches and precisions

Approach	Cluster threshold	Precisions
Euclidean clustering [11]	0.5 m	64.5%
DBSCAN [16]	Adaptive	93.7%
Depth clustering [18]	10°	39.2%
Run clustering [19]	Params _{SLR}	51.7%
Online learning [20]	Adaptive	89.8%
Our (EDBSCAN)	Adaptive	93.9%
Our (EOPTICS)	Adaptive	96.9%

and got 93.7% accuracy in-person tracking during walking. This research work proposed two novel techniques EDBSCAN and EOPTICS, with an adaptive threshold selection mechanism and found a maximum of 93.3% and 96.9% accuracy, respectively, with the same datasets.

5. Conclusion

This research introduced an augmented comprehension for person tracking with a LiDAR sensor through the exclusive introduction of density-based clustering with its prior modifications. We were applying new parameters and its collaborative expansions made these algorithms sophisticated, especially for LiDAR data. While video and multisensor based human tracking were improved in the time being, they have a precautionous consideration of privacy. Eventually, these improved tracking systems need higher computations and costs to be implemented. We considered all these constraints and compared our method with other state-of-the-art technologies. Low installation and computational costs, runtime processing, and adaptive parameter selection of algorithms made our system robust. A well-prepared own dataset made our system more accessible to others.

We intend to apply this to a mobile robot that can detect a person independently in real-time. Some commercial applications also can be made with this approach, along with the caregiving facility. We intend to develop a LiDAR-based gait recognition system. In

the future, a DNN architecture can be applied to overcome the shortcomings of many sample's learning and training.

References

- (1) Brubaker MA, Sigal L, Fleet DJ. Video-based people tracking. In *Handbook of Ambient Intelligence and Smart Environments*. Nakashima H, Aghajan H, Augusto JC (eds). Springer: Boston, MA; 2010.
- (2) Devarakota PR, Castillo-Franco M, Ginhoux R, Mirbach B, Kater S, Ottersten B. 3-D-skeleton-based head detection and tracking using range images. *IEEE Transactions on Vehicular Technology* 2009; **58**(8):4064–4077.
- (3) McLaughlin N, Rincon JMD and Miller P. Recurrent convolutional network for video-based person re-identification. Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, 2016; 1325–1334.
- (4) Roberto Henschel, Timo von Marcard and Bodo Rosenhahn. Simultaneous identification and tracking of multiple people using video and IMUs. Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshops, 2019.
- (5) Köppe E, Bartholmai M, Liers A and Schiller J. Radio-based multi-sensor system for person tracking and indoor positioning. Proceedings of ninth Workshop on Positioning, Navigation and Communication, Dresden, 2012; 180–186.
- (6) Gai W, Qi M, Ma M, Wang L, Yang C, Liu J, Bian Y, de Melo G, Liu S, Meng X. Employing shadows for multi-person tracking based on a single RGB-D camera. *Sensors* 2020; **20**(4):1056.
- (7) Christoudias CM, Urtasun R and Darrell T. Unsupervised feature selection via distributed coding for multi-view object recognition. Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Anchorage, AK; 2008, 1–8.
- (8) Kumar S, Marks TK and Jones M. Improving person tracking using an inexpensive thermal infrared sensor. Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshops, Columbus, OH, 2014; 217–224.
- (9) Benedek C, Gálai B, Nagy B, Jankó Z. Lidar-based gait analysis and activity recognition in a 4D surveillance system. *IEEE Transactions on Circuits and Systems for Video Technology* 2018; **28**(1):101–113.
- (10) M Ester, H-P Kriegel, J Sander and X Xu. A density-based algorithm for discovering clusters in large spatial databases with

- noise. Proceedings of Second International Conference on Knowledge Discovery and Data Mining (KDD'96), AAAI Press, 1996; 226–231.
- (11) Rusu RB and Cousins S. 3D is here: Point cloud library (PCL). Proceedings of IEEE International Conference on Robotics and Automation, Shanghai, 2011; 1–4.
 - (12) Wang C, Ji M, Wang J, Wen W, Li T, Sun Y. An improved DBSCAN method for LiDAR data segmentation with automatic eps estimation. *Sensors* 2019; **19**(1):172.
 - (13) Hollaus M, Wagner W, Eberhöfer C, Karel W. Accuracy of large-scale canopy heights derived from LiDAR data under operational constraints in a complex alpine environment. *ISPRS Journal of Photogrammetry and Remote Sensing* 2006; **60**(5):323–338.
 - (14) Niemeyer J, Rottensteiner F, Soergel U. Contextual classification of lidar data and building object detection in urban areas. *ISPRS Journal of Photogrammetry and Remote Sensing* 2014; **87**:152–165.
 - (15) Campello, RJGB, D Moulavi and J Sander. Density-based clustering based on hierarchical density estimates. Proceedings of Pacific-Asia Conference on Knowledge Discovery and Data Mining, Springer, Berlin, Heidelberg, 2013.
 - (16) Hasan M, Hanawa J, Goto R, Fukuda H, Kuno Y and Kobayashi Y. Tracking people using ankle-level 2D LiDAR for gait analysis. *Advances in Artificial Intelligence, Software and Systems Engineering*, AHFE 2020. Advances in Intelligent Systems and Computing, vol 1213 Springer, Cham, 2020
 - (17) Amini A, Wah TY, Saboo H. Density-based data streams clustering algorithms: A survey. *Journal of Computer Science and Technology* 2014; **29**:116–141.
 - (18) Bogoslavskyi I and Stachniss C. Fast range image-based segmentation of sparse 3d laser scans for online operation. Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), 2016; 163–169.
 - (19) Zermas D, Izzat I and Papanikolopoulos N. Fast segmentation of 3D point clouds: A paradigm on LiDAR data for autonomous vehicle applications. Proceedings of IEEE international conference on robotics and automation (ICRA) Singapore, 2017; 5067–5073
 - (20) Yan Z, Duckett T, Bellotto N. Online learning for 3D LiDAR-based human detection: Experimental analysis of point cloud clustering and classification methods. *Auton Robot* 2020; **44**:147–164.

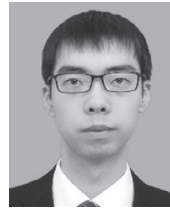
Mahmudul Hasan (Non-member) is a Ph.D. student at Human-Computer Interaction Lab, Saitama University, Japan. Besides this, he is a faculty member of the Department of Computer Science and Engineering, Comilla University, Bangladesh. His research focus is on human-robot interaction and computer vision.



Junichi Hanawa (Non-Member) is a student at the Graduate School of Science and Engineering, Saitama University. He received a B.E. degree in Information and Computer Sciences from Saitama University in 2020. His research interests include computer vision for human sensing.



Riku Goto (Non-Member) is a student at the Graduate School of Science and Engineering, Saitama University. He received a B.E. degree in Information and Computer Sciences from Saitama University in 2019. His research interests include computer vision for human sensing.



Hisato Fukuda (Non-Member) received the Ph. D. in Information and Computer Sciences from Saitama University, Saitama, Japan in 2015. Now, he is an assistant Professor at the Graduate School of Science and Engineering, Saitama University. His research interests include object recognition and human–computer interaction.



Yoshinori Kuno (Member) is an honorary Professor at Saitama University. He completed his Ph.D. in 1982 at the University of Tokyo and worked as a visiting researcher at Carnegie Mellon University, the United States, in 1988. He worked for Osaka University from 1993 to 2000 as an associate Professor. After that he joined the Department of Information and Computer Sciences, Saitama University as a Professor. His research interest includes computer vision, intelligent robotics, human–computer interaction.



Yoshinori Kobayashi (Non-Member) completed his Ph.D. at the University of Tokyo in 2007. He then joined the Department of Information and Computer Sciences, Saitama University as an Assistant Professor. Currently, he is a Professor at Saitama University. His research interests include computer vision for human sensing and its application for HCI and HRI.

