

Temporal Convolutional Networks for Multiperson Activity Recognition Using a 2-D LIDAR

Fei Luo^{ID}, Stefan Poslad^{ID}, and Eliane Bodanese^{ID}

Abstract—Motion trajectories contain rich information about human activities. We propose to use a 2-D LIDAR to perform multiple people activity recognition simultaneously by classifying their trajectories. We clustered raw LIDAR data and classified the clusters into human and nonhuman classes in order to recognize humans in a scenario. For the clusters of humans, we implemented the Kalman filter to track their trajectories which are further segmented and labeled with corresponding activities. We introduced spatial transformation and Gaussian noise for trajectory augmentation in order to overcome the problem of unbalanced classes and boost the performance of human activity recognition (HAR). Finally, we built two neural networks, including a long short-term memory (LSTM) network and a temporal convolutional network (TCN) to classify trajectory samples into 15 activity classes collected from a kitchen. The proposed TCN achieved the best result of 99.49% in overall accuracy. In comparison, the TCN is slightly superior to the LSTM network. Both the TCN and the LSTM network outperform the hidden Markov model (HMM), dynamic time warping (DTW), and support vector machine (SVM) with a wide margin. Our approach achieves a higher activity recognition accuracy than the related work.

Index Terms—Human activity recognition (HAR), localization, long short-term memory (LSTM), people tracking, temporal convolutional network (TCN), trajectory classification.

I. INTRODUCTION

HUMAN activity recognition (HAR) has a wide range of applications in surveillance, healthcare, smart home, intelligent control, etc. For example, doctors and dietitians can provide advice remotely for patients and customers based on their daily dietary activities [1]. People can use gesture recognition [2], [3] to contactlessly interact with electronic devices, such as TV, computers, smart glasses, etc. Activity recognition can be used to detect abnormal behavior patterns in many surveillance tasks [4], for instance, human falling detection can be very helpful in providing immediate medical assistance for the elderly [5].

A range of sensors have been applied for HAR. Video-based activity recognition has been well developed. Nunez *et al.* [6] extracted human skeletal outlines from videos to perform activity identification and gesture recognition. Ibrahim *et al.* [7]

derived a temporal representation of person-level actions from sports videos and combined the representation of individual people to recognize group activities. However, video-based systems can be affected by insufficient illumination and privacy concerns. Wearable inertial sensors, including accelerometer, gyroscope, and magnetometer, have increasingly been applied to HAR as they are fairly ubiquitous, embedded in smartphones, smartwatches, and sports bracelets. Jung *et al.* [8] used air-pressure sensors to measure the change of the air pressure resulted from muscular activities in order to recognize gestures. Heterogeneous wearable sensors have been fused in order to recognize complex human activities with more accurate results. Qi *et al.* [9] proposed a two-layer recognition framework to classify gym physical activities by fusing accelerometers and electrocardiograms. However, the users may forget to wear wearable sensors [10] or feel uncomfortable to wear them [11]. Consequently, wireless sensing has been increasingly used instead. For example, Wi-Fi [12] and Radar [13] also have been used in HAR. Wang *et al.* [14] used the Wi-Fi channel state information (CSI) to perform fall detection. Ren *et al.* [15] used the range–time–frequency information obtained from a Doppler radar to perform subject localization and heartbeat detection. Compared to cameras and wearable sensors, Wi-Fi and Radar-based techniques are more suitable for HAR indoors, which has less interference than outdoors.

Location acquisition systems have been widely applied in our life and several techniques have been used, such as GPS, Radar, LIDAR, etc. These systems generate a large amount of location data, and in the past years, they have been applied in navigation, route planning, and mapping. Recently, more studies attempt to infer semantic information from the trajectories formed by the locations of people, in order to provide a higher level of services, for example, for a personalized recommendation, smarter home or more intelligent human-like robot interaction. Although there are many localization techniques, none of these can be applied ubiquitously for all purposes. GPS hardly works indoors as its signals cannot penetrate walls. Radars may suffer interference from the ground surrounding and multipath effects. Bluetooth and radio-frequency identification (RFID) can perform indoor localization, however, due to their short-range sensing capability, they generally require larger deployments that may become difficult to maintain. As a surveying and mapping technique, LIDAR has been increasingly used in self-driving vehicles and robots due to its high localization accuracy, good real-time performance, and easy deployment. LIDAR also can be applied both indoors and outdoors. As 2-D LIDARs are more affordable than 3-D LIDARs,

Manuscript received December 10, 2019; revised January 23, 2020 and February 27, 2020; accepted March 27, 2020. Date of publication March 31, 2020; date of current version August 12, 2020. This work was supported by the Queen Mary–China Scholarship Council Co-Funded Scholarships. (Corresponding author: Fei Luo.)

Fei Luo is with the School of Geospatial Engineering and Science, Sun Yat-sen University, Zhuhai, China (e-mail: f.luo@qmul.ac.uk).

Stefan Poslad and Eliane Bodanese are with the School of Electronic Engineering and Computer Science, Queen Mary University of London, London E1 4NS, U.K.

Digital Object Identifier 10.1109/IJOT.2020.2984544

this article proposes to use a 2-D LIDAR to perform multiple people activity recognition indoors.

A trajectory collected by using a LIDAR is a sequence of coordinates. A trajectory is a time series that contains both spatial and temporal information. Recurrent neural networks (RNNs) are able to model long-term dependencies in time series by propagating information through their deterministic hidden state. As a variant of RNN, long short-term memory (LSTM) further uses three gates to regulate the flow of information in and out of each node, which overcomes the exploding and vanishing gradient problem that exists in conventional RNNs. The temporal convolutional networks (TCNs) leverage large receptive fields by stacking many dilated convolutions, allowing them to model even longer time scales up to the entire sequence length [16]. Both LSTM and TCNs have achieved outstanding results in many areas, such as sequential image classification, audio classification, language modeling, etc. In this article, we build an LSTM network and a TCN to classify human trajectories into predefined activities and make a comparison between them.

It is more challenging to perform HAR using wireless sensing techniques than using wearable sensors. For wearable sensor-based HAR, each person wears sensor devices, the sensor data are collected from each specific person. However, wireless sensing techniques, including LIDAR, Radar, and Wi-Fi monitor a scenario that contains both humans and nonhuman objects, such as walls, tables, animals, etc. It is necessary to differentiate humans from nonhuman objects before performing HAR. Another challenge is that multiple people can appear in the detection area of wireless sensors. For recognizing the activities of different people, it is necessary to track each person separately. LIDAR can achieve centimeter-level localization accuracy, by using this characteristic of LIDARs, it is possible to separate multiple people in the spatial dimension. LIDARs also have a very fast response time that makes them suitable for real-time tracking. These reasons drove us to choose a 2-D LIDAR to perform HAR.

In this article, we used a 2-D LIDAR to track multiple people simultaneously and collect their trajectories for activity recognition. By using a very short sliding window (2.5 s) for trajectory segmentation, our approach is able to achieve nearly real-time HAR. We applied spatial transformation and Gaussian noise for trajectory augmentation in order to overcome the problem of unbalanced classes. Finally, we built two neural networks, an LSTM network and a TCN, and compared their performance. Our proposed approach is able to deliver a very high classification accuracy of indoor activities in nearly real-time, which is essential to make feasible real-time monitoring applications.

The remainder of this article is organized as follows. Section II presents the related work for HAR from three aspects: 1) sensors; 2) machine learning; and 3) trajectory-based HAR. Section III illustrates the detailed methodologies, including data processing, point clustering, multiuser tracking, trajectory segmentation and augmentation, and our proposed LSTM and TCN networks. In Section IV, we implement our method in an indoor scenario. We evaluate our proposed

approach and present the results in Section V. Finally, we draw the conclusion in Section VI.

II. RELATED WORK

HAR has been widely researched by using a diversity of sensing devices and methods. In this section, we will review some related work with respect to types of sensors, machine learning algorithms, and trajectory-based HAR.

A. Sensors

A variety of sensors have been applied in HAR, such as cameras, RFIDs, Wi-Fi, etc. Two kinds of techniques emerged from these types of sensors: 1) device-bound and 2) device-free techniques. The device-bound techniques require users to carry wearable sensors, such as electrocardiogram sensors, accelerometers, Bluetooth, or Wi-Fi receivers. As many wearable sensors are attached to the human body, they have great advantages in acquiring physiological characteristics [17], including body temperature, heart rate, and blood pressure, which are helpful for healthcare monitoring. The device-free techniques do not require users to wear any sensor. The sensing of activities is performed remotely by using, for example, cameras, Radars, LIDARs, etc. The device-free techniques have been attracting more attention as they are more convenient and require no effort or attachments from users. However, each sensor has its own advantages and limitations. Cameras are widely deployed for HAR because videos and images are visual forms that can be easily perceived by our eyes. However, most cameras suffer from insufficient illumination and have a narrow visual range that makes them unsuitable for larger areas. It is also privacy intrusive because people's faces may be exposed. RFID tags [18] and Bluetooth beacons [19] usually require a large deployment and maintenance because they are very short range. They may need users to carry an RFID reader or a Bluetooth receiver to read the tags or beacons embedded in specific parts of the physical environment. Radars can provide micro-Doppler signatures generated from human activities [13]. Radars do not generate privacy concerns because personal identifiable features cannot be obtained with Radar detection, and they can penetrate walls, cloth, trees, etc. Wi-Fi devices can measure the CSI or the received signal strength indicator (RSSI) to perform activity recognition. They have similar merits as the Radar devices. However, both Wi-Fi and Radar devices may present a weak anti-interference capability and suffer from multipath effects. GPS and LIDAR have been used in HAR through the analysis of patterns of location sequences of humans. As GPS is hardly received indoors, it is usually used outdoors within large and open spaces. LIDAR is a surveying method that measures the distance to a target by illuminating the target with pulsed laser light and measuring the reflected pulses. It has been widely applied for object localization, tracking, and mapping. Currently, few researchers have applied LIDAR to HAR due to its expensive cost.

B. Machine Learning Algorithms

Machine learning has been widely applied in HAR to classify different activities. Dynamic time warping (DTW) is one

of the most used algorithms in HAR [20], [21]. It measures the similarity between two temporal sequences to find the optimum distance between time series [22]. As almost all data captured by sensors in HAR are time series, DTW has innate advantages in processing them. The hidden Markov model (HMM) is also favored in HAR [23], [24]. It is a Markov chain with both hidden and observable stochastic processes. In HAR, observable components are the sensor signals while hidden elements are users' activities [25]. Other algorithms, such as support vector machine (SVM) [26], k -nearest neighbor (kNN) [27], and random forest (RF) [28] also have been applied in HAR. Since deep learning has achieved very high performance in many fields, it is also increasingly applied to HAR. Kamel *et al.* [29] built a convolutional neural network (CNN) to classify RGB-D data set in order to recognize human actions. Tang *et al.* [30] proposed a coherence-constrained graph LSTM to recognize group activity by modeling the relevant motions of individuals while suppressing the irrelevant motions. Guo *et al.* [31] proposed a feature fusion method, multiview Cauchy estimator feature embedding, to fuse the data of Kinect and inertial sensors for human action recognition. Lea *et al.* [32] used TCNs to build an encoder-decoder model for human action segmentation and recognition upon videos. Luo *et al.* [1] used a CNN to classify Radar frequency spectrograms for an indoor HAR comprising of 15 different activities. Some of the above algorithms have also been used in trajectory-based HAR as presented in Section II-C.

C. Trajectory-Based HAR

Most trajectory-based human activity detection is achieved by using cameras. Siang and Khor [33] built a DTW model to cluster similar trajectories in a video surveillance system for suspicious behavior detection. Liang *et al.* [34] proposed an LSTM model to perform trajectory and activity prediction from videos. Wang *et al.* [35] proposed to detect abnormal behaviors through trajectory analysis and anomaly modeling based upon a camera sensor network. Gao and Sun [36] proposed a hierarchical Dirichlet process HMM (HDP-HMM) to model the transitions of different motions in video sequences. However, besides the shortcomings of small visual scope and privacy intrusion, trajectories extracted from a video are heavily dependent on the azimuth and inclination of the camera, and the coordinates of trajectories are hardly integrated into a global coordinate system [4]. GPS as the most popular outdoor localization technique also has been used for trajectory-based activity recognition. Fuse and Kamiya [37] built a hierarchical Dirichlet process HMM upon GPS trajectories to detect anomalies of human motion in dynamic traffic control. Huang *et al.* [38] proposed an automatic activity detection method using Points of Interest (PoIs') spatial-temporal attractiveness to identify activity locations as well as durations from raw GPS trajectories. Huang *et al.* [39] proposed a cost-sensitive approach for activity recognition from GPS-logs by measuring the importance of each activity from spatial and temporal perspectives. Martin *et al.* [40] modeled the human mobility behavior using GPS trajectories to predict the purpose of a user's visit at a certain location.

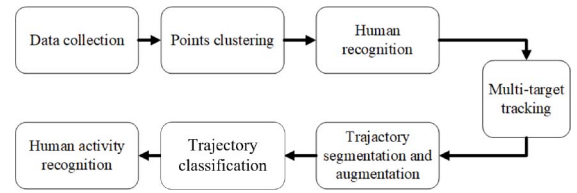


Fig. 1. Workflow of LIDAR-based HAR.

However, GPS requires users to carry a GPS receiver and it is very difficult to receive GPS signals indoors. Currently, there is little research in trajectory-based HAR using LIDAR. Panangadan *et al.* [41] used a 2-D laser to investigate interactions between persons and detect anomalies in three different environments: 1) open layout laboratory; 2) corridor; and 3) an outdoor courtyard. Ma *et al.* [42] built a seq2seq model to model the location-driven sequences of a user in order to recognize human activity in a kitchen. They also used a 2-D LIDAR. However, they did not differentiate human targets and obstacles, and the system recognized activities of only one person, which is fine for single people occupancy, but not for a family home. In our research, we further classify human and nonhuman targets and perform continuous tracking to obtain trajectories of multiple people.

III. METHODOLOGY

The main steps of the LIDAR-based HAR are demonstrated in Fig. 1. First, we use a 2-D LIDAR to collect a set of points, which are reflected by walls, tables, humans, etc. This set of points are grouped into different clusters that represent different objects in a scene. We further perform human recognition upon these point clusters by using an RF to classify geometric features extracted from each cluster. We use the Kalman filter to perform continuous tracking of multiple people. The trajectories obtained from the tracking are further segmented and tagged with our predefined activity labels. Trajectory augmentation is implemented to enrich the samples and overcome the problem of unbalanced classes. Finally, we build an LSTM network and a TCN to perform trajectory classification. After training, both two networks are able to perform HAR online. The details of each step are described in the following sections.

A. Point Clustering

The raw data of a LIDAR are a sequence of polar coordinates and each coordinate is noted as (r, θ) , where r is the distance to LIDAR and θ is the angle. For facilitating subsequent processing, we transformed these coordinates from a polar coordinate system to a plane coordinate system. Then, the data at each timestep can be described as a sequence $S = \{p_1, p_2, \dots, p_n\}$, where n is the total number of points.

In order to differentiate objects, it is necessary to perform clustering to group the points in S into different clusters. We applied an algorithm called a density-based spatial clustering algorithm (DBSCAN) to perform the clustering. The DBSCAN is based on a threshold for a number of neighbors, $minPts$, within a radius ϵ . A point with the neighbor count greater than or equal to $minPts$ within ϵ is identified as a

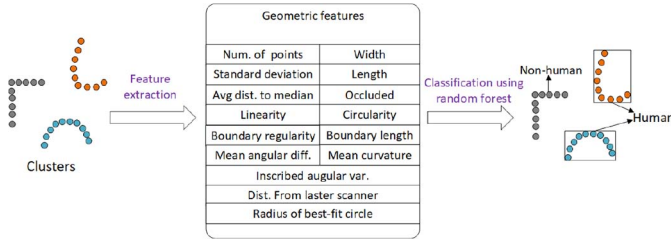


Fig. 2. Human recognition.

core point. A border point has the number of neighbors that is less than $minPts$ but it belongs to the ε -neighborhood of other points. If a point is neither a core nor a border point, then it is called a noise point. In order to understand how DBSCAN works, three terms are defined.

- 1) *Direct Density Reachable*: A point A is directly density reachable from another point B if A is in ε -neighborhood of B and B is a core point.
- 2) *Density Reachable*: A point A is density reachable from B if there are a set of core points leading from B to A .
- 3) *Density Connected*: Two points A and B are density connected if there is a core point C , such that both A and B are density reachable from C .

The DBSCAN can be abstracted into the following steps.

- 1) Find all neighbor points within ε of every point, and each point with more than $minPts$ neighbors within ε are marked as a core point.
- 2) For each core point if it is not already assigned to a cluster, create a new cluster. Find recursively all its density-connected points and assign them to the same cluster as the core point.
- 3) Iterate through the remaining unvisited points in the data set. Those points that do not belong to any cluster are marked as noise.

Compared to the other clustering algorithms, DBSCAN does not need to specify the number of clusters, it is able to discover arbitrarily shaped clusters, and it is robust to noise [43].

B. Human Recognition

The clusters obtained after point clustering represent many different objects. It is necessary to recognize which cluster represents humans in order to track its trajectory. The clusters that represent humans have different geometric characteristics from clusters of nonhuman targets. The geometric feature can be defined in terms of linearity, circularity, angularity, etc. As Leigh *et al.* did in [44], we extracted 15 geometric features (shown in Fig. 2) from each cluster. Based on these features, we implement an RF to classify human and nonhuman targets. RF is a tree-based ensemble classifier which is a meta estimator that fits several decision trees on various subcategories of data. This algorithm is used for both regression and classification. In general, the number of trees determines the robustness of the forest. We collect positive samples (human) from a clear and open space and negative samples (nonhuman) from various indoor spaces without people. The total number of samples was 12000 consisting of 6000 positive samples and 6000 negative samples. One benefit of RF is that it can

measure the confidence level in the classification by aggregating the classification output of each individual tree. Thus, rather than just using the predicted labels, we consider the confidence level generated from the RF classifier to eliminate the noise and increase reliability.

C. Multiple People Tracking

In order to obtain the trajectories of human clusters, it is necessary to perform continuous target tracking. The Kalman filter is one of the most popular tracking algorithms due to its efficiency and accuracy [45]. It is an optimal recursive data processing, which uses a series of measurements observed over time to produce an estimate of the desired variables and finds the optimal state with the smallest possible variance error. It contains two steps: 1) prediction and 2) correction. In the prediction step, the state is predicted with the dynamic model, while in the correction step, the state is corrected with the observation model.

The process and measurement equations for the Kalman filter are given as follows:

$$\begin{aligned} x_k &= Ax_{k-1} + Bu_k + w_{k-1} \\ z_k &= Hx_k + v_k \end{aligned} \quad (1)$$

where k is the discrete time, x_k is the state vector, z_k is the observation vector, and A and H are the transition matrix and the observation matrix, respectively. B_k is the control-input model which is applied to the control vector u_k . w_{k-1} and v_k are the Gaussian random variables with zero mean, so their probability distributions are $p(w) \sim N(0, Q)$ and $p(v) \sim N(0, R)$, where the covariance matrices Q and R are referred to as transition noise covariance matrix and observation noise covariance matrix.

The prediction stage for the Kalman filter is as follows:

$$\begin{aligned} \hat{x}_k^- &= A\hat{x}_{k-1} + Bu_k \\ P_k^- &= AP_{k-1}A^T + Q \end{aligned} \quad (2)$$

a priori estimate of state $\hat{x}_k^-$ and covariance error P_k^- is obtained for the next time step k .

The correction stage for the Kalman filter is as follows:

$$\begin{aligned} K_k &= P_k^- H^T (HP_k^- H^T + R)^{-1} \\ \hat{x}_k &= \hat{x}_k^- + K_k(z_k - H\hat{x}_k^-) \\ P_k &= (1 - K_k H)P_k^- \end{aligned} \quad (3)$$

K_k is the Kalman gain which is computed by the above equations. After that, a posterior state estimate $\hat{x}_k^-$ and a posterior error estimate P_k is computed by the measurement z_k . The prediction and correction equations are calculated recursively with the previous posterior estimates to predict new prior estimates.

In a tracking system, the state vector is $X = [x, y, v_x, v_y, a_x, a_y]^T$, where (x, y) , (v_x, v_y) , and (a_x, a_y) represent the position, velocity, and acceleration, respectively. The observation vector is $Z = (x', y')$.

D. Trajectory Segmentation and Augmentation

A trajectory can be the result of the location displacement when one or a sequence of activities is performed.

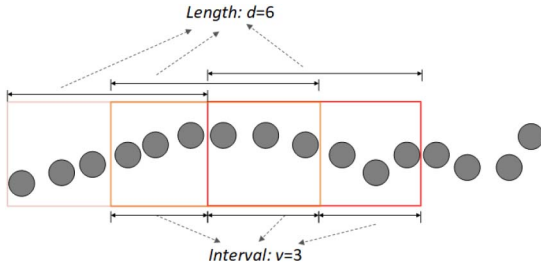


Fig. 3. Trajectory segmentation.

For differentiating activities, we split a trajectory into segments that each segment represents one activity. Several ways have been used to perform trajectory segmentation, such as distance-based segmentation, time-based segmentation, and a number of points-based segmentation. They split a trajectory into segments with the same distance, time, and number of points, respectively. In our research, we implement the number of points-based segmentation. We use a sliding window with a length of d to perform segmentation at the interval of v . For instance, Fig. 3 shows a sliding window with a length of six (location points) and the sliding interval of three (location points).

In our daily life, the frequency and duration of different activities are different. For example, the activity of “having a meal” is mainly performed three times a day but the activity such as “washing hands” can be performed many times, while the time consumed by having a meal is much longer than washing hands. This leads the problem of data sparsity that some activities have many samples but others have few. To overcome this problem, we applied trajectory augmentation by using spatial transformation. In machine learning, data augmentation is used to create more training samples through different ways of processing or a combination of multiple processing upon the original samples. Data augmentation can increase the number of samples and boost the performance of deep learning. In this article, we applied spatial transformation upon the raw trajectories to perform trajectory augmentation because the trajectories are spatial data. The methods are described as follows.

- 1) *Translation*: All points are translated to new positions by adding offsets T_x and T_y to x and y , respectively

$$\begin{bmatrix} x' & y' & 1 \end{bmatrix} = \begin{bmatrix} x & y & 1 \end{bmatrix} \cdot \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ T_x & T_y & 1 \end{bmatrix}. \quad (4)$$

- 2) *Rotation*: All points in the 2-D plane are rotated around the origin through the counterclockwise angle θ

$$\begin{bmatrix} x' & y' & 1 \end{bmatrix} = \begin{bmatrix} x & y & 1 \end{bmatrix} \cdot \begin{bmatrix} \cos \theta & \sin \theta & 0 \\ -\sin \theta & \cos \theta & 0 \\ 0 & 0 & 1 \end{bmatrix}. \quad (5)$$

- 3) *Scale*: All points are scaled by applying the scale factors S_x and S_y to the x and y coordinates, respectively

$$\begin{bmatrix} x' & y' & 1 \end{bmatrix} = \begin{bmatrix} x & y & 1 \end{bmatrix} \cdot \begin{bmatrix} S_x & 0 & 0 \\ 0 & S_y & 0 \\ 0 & 0 & 1 \end{bmatrix}. \quad (6)$$

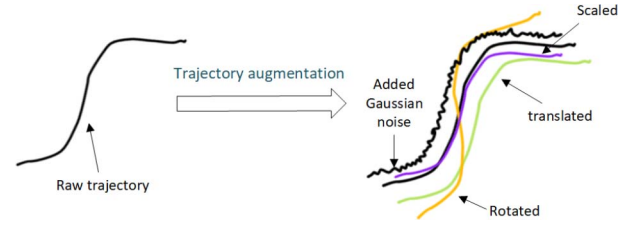


Fig. 4. Trajectory augmentation.

In addition to spatial transformation, we also added the Gaussian noise to the raw trajectories. The process of trajectory augmentation is shown in Fig. 4. It is worthy to note that it is required to control the range of spatial transformation to avoid changing the semantic information of trajectories.

E. Trajectory Classification

1) *Long Short-Term Memory*: LSTM is a variant of RNNs that have been successfully applied in sequential learning. In RNNs, the gradient increases rapidly or decays exponentially over time. For overcoming this “exploding and vanishing gradient” problem, LSTM uses a few gates to control the passing of the information along the sequence and thus can learn long-range dependencies. A typical LSTM unit contains an input gate i_t , a forget gate f_t , a cell c_t , an output gate o_t , and an output response h_t . The input gate controls the extent to which a new value flows into the cell, the forget gate controls the extent to which a value remains in the cell, and the output gate controls the extent to which the value in the cell is used to compute the output activation of the LSTM unit. The recursive computation of an LSTM unit is

$$\begin{aligned} i_t &= \delta(W_{xi}x_t + W_{hi}h_{t-1} + W_{ci}c_{t-1} + b_i) \\ f_t &= \delta(W_{xf}x_t + W_{hf}h_{t-1} + W_{cf}c_{t-1} + b_f) \\ c_t &= f_t \odot c_{t-1} + i_t \odot \tanh(W_{xc}x_t + W_{hc}h_{t-1} + b_c) \\ o_t &= \delta(W_{xo}x_t + W_{ho}h_{t-1} + W_{co}c_{t-1} + b_o) \\ h_t &= \delta \odot \tanh(c_t) \end{aligned} \quad (7)$$

where $\odot$ denotes the elementwise product, $\delta(x)$ is the sigmoid function defined as $\delta(x) = 1/(1 + e^{-x})$, $W_{\alpha\beta}$ is the weight matrix between α and β (e.g., W_{xi} is the weight matrix from the input x_t to the input gates i_t), and b_β denotes the bias term of β with $\beta \in i, f, c, o$.

As shown in Fig. 5, we built an LSTM network that consists of two LSTM layers and one fully connected layer. The segmented trajectories were input into the network. As each point in a trajectory has two values (x, y), the input shape of the LSTM network is $(t, 2)$, where t is the length of a trajectory. Both two LSTM layers have 120 units and the fully connected layer has 200 hidden units. The size of the output layer is determined by the number of activities that we are going to recognize.

2) *Temporal Convolutional Network*: TCN is a class of time-series neural network models that capture long-range patterns using a hierarchy of temporal convolutional filters [32]. TCNs integrate dilated casual convolutions and the residual block structure to expand the receptive field and increase the

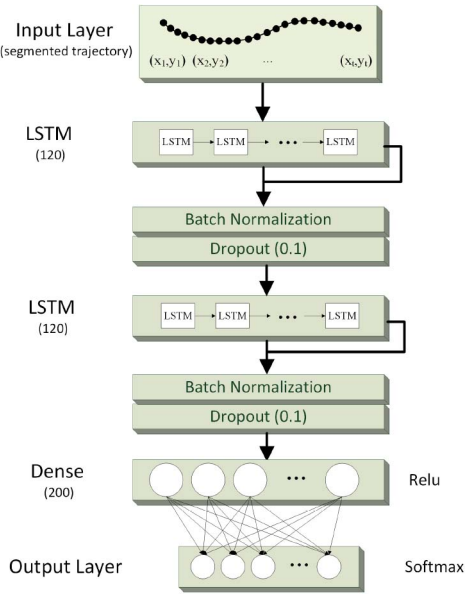


Fig. 5. Structure of our proposed LSTM network.

depth. In dilated casual convolutions, an output at time t is convolved only with the elements from time t and earlier in the previous layer by using dilated convolution to enable an exponentially large receptive field [46]. For a 1-D sequence input $x \in \mathbb{R}^n$ and a filter $f: 0, \dots, k-1 \rightarrow \mathbb{R}$, the dilated convolution operation F on the element s of the sequence is defined as [46]

$$F(s) = (x * f)(s) = \sum_{i=0}^{k-1} f(i) \cdot x_{s-d \cdot i} \quad (8)$$

where d is the dilation factor, k is the filter size, and $s - d \cdot i$ accounts for the direction of the past. With dilated casual convolutions, the receptive field of the TCN can be increased by using larger filter sizes k and increasing the dilation factor d . Fig. 6(a) presents a dilated casual convolution with the dilation factors $d = 1, 2, 4$ and a filter size $k = 2$.

A TCN consists of several residual blocks. Each residual block contains a series of transformations τ . As shown in Fig. 6(b), a residual block consists of two sets of layers, including a dilated casual convolution layer, a weight normalization layer, a ReLU activation layer, and a dropout layer. The output of each residual block is the sum of the output of these transformations and the input x

$$o = \text{Activation}(x + \tau(x)). \quad (9)$$

This is also called skip connection. The residual blocks avoid the vanishing gradient problem by carrying gradient throughout the extent of a very deep network.

In this article, we built a TCN that consists of three residual blocks (“Block 1,” “Block 2,” and “Block 3”) as shown in Fig. 6(c). After hyperparameter tuning, the kernel sizes for these three blocks were set as 5, and the dilation rates of them were 1, 2, and 4, respectively. Block 1 has 75 filters, Block 2 has 150 filters, and Block 3 has 300 filters.

In the training process, both the LSTM network and the TCN network were trained to minimize an objective function

in terms of the parameters of the network. For activity recognition, let C be the number of activities, the following cross-entropy loss function is often used:

$$E_y(y') = - \sum_{i=1}^N y_i \cdot \log(y'_i) \quad (10)$$

where E is the loss function evaluated over N samples, y_i is the original label of the i th sample, and y'_i is the class score maps of the sample i calculated using a softmax activation function

$$y_j = \exp(x_j) / \left(\sum_{c=1}^C \exp(x_c) \right) \quad (11)$$

where y is the softmax score and x is the output layer containing unnormalized class scores.

In the training process, we applied dropout to perform regularization that prevents overfitting. The term “dropout” refers to dropping out a part of the units in a neural network. By avoiding training all nodes on all training data, dropout decreases overfitting. We initialized dropout with the rate of 0.1 for the LSTM network and 0.25 for the TCN. In the LSTM network, batch normalization was applied after each LSTM layer. Batch normalization normalizes the output of a previous activation layer by subtracting the batch mean and dividing by the batch standard deviation. Except for preventing overfitting, batch normalization also can accelerate the training. The optimizer used for both networks is Adam whose learning rate was initialized as 0.0003.

IV. EXPERIMENTAL SETUP

In this section, we present the LIDAR and our experiments. The LIDAR we used is a UST-10LX as shown in Fig. 7(a). Its error in localization accuracy is within 40 mm. It has a maximum detection distance of 30 m and a scan angle of 270° . Its angular resolution is 0.25° . More information about UST-10LX can be found in [47]. We made the experiments in a kitchen scenario located in a university campus. Activities in the kitchen are directly related to dietary health. The kitchen scene context-based activity recognition can be a useful method for diet controlling and dietary treatment, also helpful for developing a smart kitchen as a part of a smart home. Fig. 7(b) presents the bird’s-eye view of the 3-D model of the kitchen.

In our experiments, the LIDAR is placed at 1-m high and close to the up-left corner of the kitchen. We use a sampling frequency of 10 Hz (10 scans per second) to collect the data and further process the collected data using the methods described in Section III as shown in Fig. 8. The raw Lidar data [Fig. 8(a)] consist of the points reflected from obstacles, walls, and humans. After clustering using DBSCAN, the points are clustered into different groups that are shown with different colors in Fig. 8(b). We perform human recognition by classifying 15 geometric features extracted from these clusters using RF with 25 trees. Finally, we build a Kalman filter for each human target in order to obtain their trajectories.

We monitor the kitchen using the UST-10LX for a whole day and plot a density map of human trajectories to present

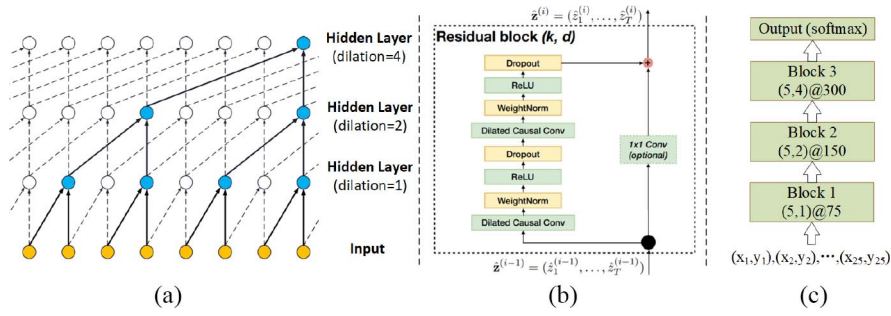


Fig. 6. Architecture of our proposed TCN. (a) Dilated causal convolution. (b) Structure of residual blocks in TCN. (3) Structure of our TCN.



Fig. 7. (a) UST-10LX LIDAR. (b) Kitchen scenario.



Fig. 9. Density map of human trajectory in the kitchen.

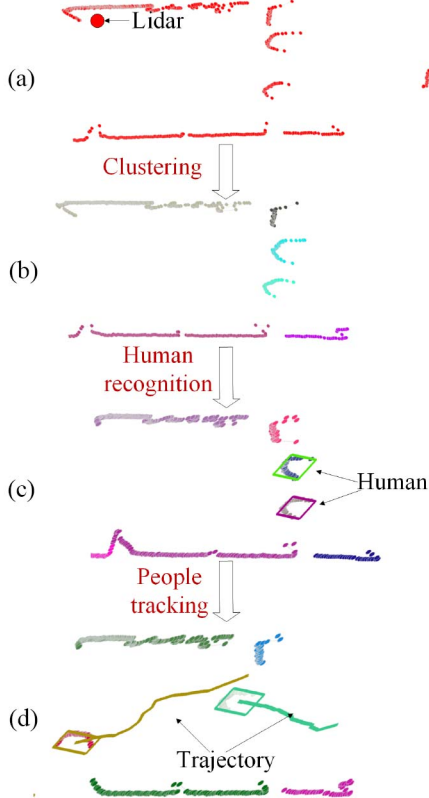


Fig. 8. LIDAR data processing.

the spatial distribution of their daily activities. As shown in Fig. 9, human trajectories are mainly concentrated around the sink, the oven, the chair, and the door in the kitchen. These four areas can be seen as four stay areas that people usually stop for some amount of time. For example, people usually stay around the sink to wash their cooking utensils, stay by the oven for cooking, and sit on the chair for eating. Based on this, we predefined 15 activities including: “get in,” “get

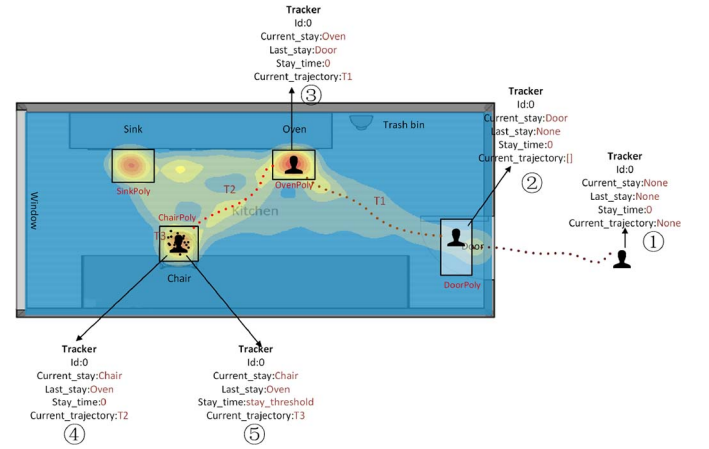


Fig. 10. Process of trajectory collection.

out,” “from sink to door,” “from chair to sink,” “from oven to chair,” “from sink to chair,” “washing,” “from door to sink,” “cooking,” “from oven to door,” “sitting,” “from oven to sink,” “from chair to oven,” “from door to oven,” and “from sink to oven.”

The trajectories of these activities (except get in and get out) begin or end at the stay areas. Based on this, we define a tracker for each person in the detection range. The properties of a tracker include “id,” “current stay,” “last stay,” “stay time,” and “current trajectory” as shown in Fig. 10. The trajectory collection process for each person is as follows.

- 1) Initialize a tracker for each person in the detection area and set the tracker’s current stay and last stay as none.
- 2) Update the tracker at each scanning frame of the LIDAR and judge whether the current location of the tracker is within a stay area. If it is within a certain stay area, then set its current stay as the name of the stay area,

and initialize its current trajectory as a point list to store the trajectory of the tracker.

- 3) When the tracker reaches a new stay area, then set its last stay as the value of current stay and its current stay as the name of this new stay area. Save its current trajectory as a trajectory labeled “from last stay to current stay,” then clear its current trajectory to empty for storing the next trajectory of the tracker.
- 4) If the tracker does not get to a new stay area but stays in the same stay area for some amount of time and the stay time is beyond the threshold (80 frames used in this article), then save its current trajectory and label it with cooking if the stay area is “OvenPoly,” label it with sitting if the stay area is “ChairPoly,” and label it with “washing” if the stay area is “SinkPoly” as shown in Fig. 10. After this, set its current trajectory as empty to store the next trajectory of the tracker.
- 5) Repeat steps 3 and 4 until the tracker leaves the detection range of the LIDAR.

The trajectories of get in and get out do not begin or end at any stay area. Therefore, we collect their trajectories manually. We start to record the trajectory of participants before they got in, and stop to record after they got in, then save the trajectories and label them as get in. For get out, the collection process was similar.

The trajectories that begin or end at the stay areas were recorded and labeled automatically. The geometric relationship between trajectories and the stay areas assures that those trajectories were correctly labeled. For get in and get out, they were recorded manually. Some of them may have been mislabeled but they could be filtered out by judging their directions, as the directions of get in and get out are opposite to each other. Besides, the collected trajectory can be visually checked.

We collected trajectories of these activities and segmented them using a sliding window with a length of 25 (location points) and an interval of 10 (location points), so each trajectory has 25 points. As the sampling frequency is 10 Hz, the length of the sliding windows also can be measured by time, in our case, it is 2.5 s. We choose this length because a longer trajectory may contain more than one activity and will lead to more latency, and a shorter trajectory does not have enough information related to human activity. In the experiments, the number of people in the kitchen was varying and the maximum number of people was 4. The total time for monitoring was about 40 h. In Fig. 11, we demonstrate one trajectory sample in our data set for each activity with different colors. As can be seen, the points of the trajectories of washing, cooking, and sitting tend to cluster together because people do not move much when they perform these activities.

As shown in Fig. 12, the quantity of each type of activity before augmentation is 357 from sink to door, 367 from chair to sink, 279 from sink to chair, 1000 washing, 1000 sitting, 350 from door to oven, 353 from oven to chair, 372 from oven to door, 1000 cooking, 267 from door to sink, 270 from chair to oven, 536 from sink to oven, 656 from oven to sink, 300 get in, and 300 get out. Some activities (cooking, washing, sitting, etc.) have many samples while others (from sink to chair, from chair to oven, etc.) do not. We implement trajectory

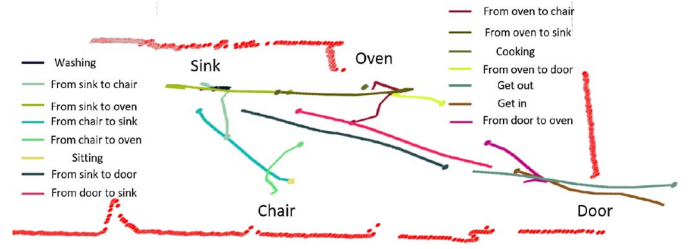


Fig. 11. Trajectory samples of human activity.

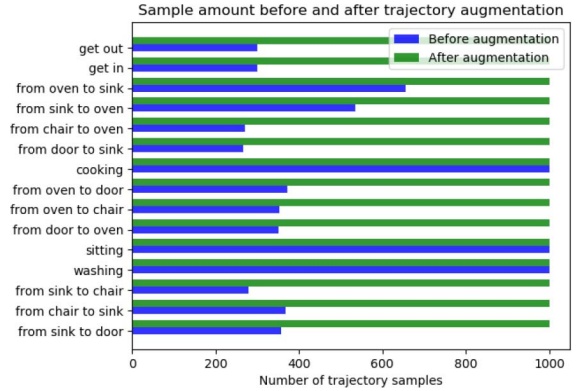


Fig. 12. Sample amount before and after trajectory augmentation.

augmentation to overcome this problem of unbalanced classes. After trajectory augmentation, we obtained 15000 samples in total and the number of samples per activity was 1000.

V. EVALUATION AND RESULTS

In this section, we evaluated our proposed neural networks in three ways, including evaluation of the test data set, comparison with the baseline, and comparison with the related work.

A. Evaluation on the Test Data Set

We split the trajectory samples into two groups, 80% of the samples were used for training, and 20% for testing. During training, we performed a hold-out validation [48], 15% of the sample data were randomly excluded from training. The testing data set was not exposed to our networks in training. The achieved overall accuracies by the LSTM network and the TCN in testing were 99.39% and 99.49%, respectively. The TCN performed slightly better than the LSTM. More importantly, the convergence speed of the TCN is significantly faster than the LSTM. As shown in Fig. 13, the lines of validation loss and accuracy of the TCN are smoother than those of the LSTM network. This is probably because TCN has a backpropagation path different from the temporal direction of the sequence that is beneficial to avoid the problem of exploding/vanishing gradients [46].

As shown in the normalized confusion matrix of Fig. 14, most of the activities are correctly classified with 100% or 99%. For from chair to sink, 1% samples have been misclassified into from chair to oven and 1% samples have been misclassified into sitting. It is because some trajectory samples from these classes begin from the chair in the kitchen

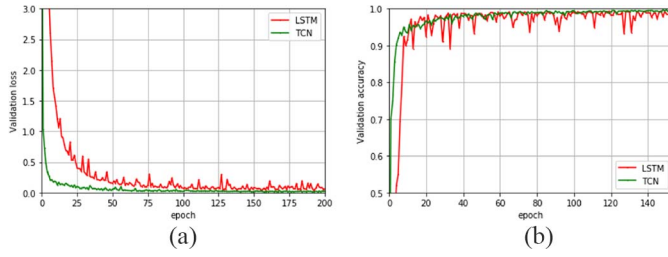


Fig. 13. Validation loss and accuracy of the proposed LSTM and TCN. (a) Validation loss. (b) Validation accuracy.

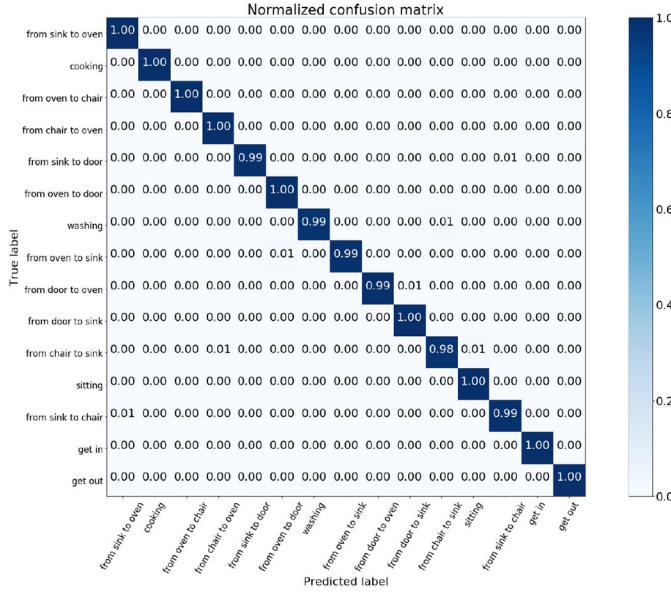


Fig. 14. Normalized confusion matrix of the TCN in testing.

TABLE I
EFFECT OF TRAJECTORY AUGMENTATION

	OA	Recall	F1
TCN with trajectory augmentation	99.49%	99.53%	99.51%
TCN without trajectory augmentation	97.96%	97.93%	97.96%
LSTM with trajectory augmentation	99.39%	99.41%	99.39%
LSTM without trajectory augmentation	97.68%	97.79%	97.65%

and have some overlaps. The same situation also happened between from door to oven and from door to sink. 1% samples of from door to oven have been misclassified into from door to sink. Also, 1% samples of from sink to door have been misclassified into from sink to chair.

For validating the effect of trajectory augmentation, we compared the networks that trained on the samples after augmentation and those trained on the raw samples. As shown in Table I, after applying our proposed trajectory augmentation, the performance of the LSTM network has been increased by about 1.7% in OA, Recall, and F1; and the performance of the TCN has been increased about 1.5% in all three metrics.

B. Comparison With Baseline

As mentioned in Section II, both HMM and DTW have been frequently applied to trajectory-based activity recognition in the related work. SVM is a widely used method in

TABLE II
COMPARISON OF THE CLASSIFICATION MODELS

	OA	Recall	F1
TCN	99.49%	99.53%	99.51%
LSTM	99.39%	99.41%	99.39%
SVM	95.88%	95.76%	96.69%
HMM	85%	86.7%	85.5%
DTW	90.9%	91.56%	91.71%

classification problems. We used these three algorithms as the baseline to make a comparison with our networks.

The SVM implemented in this article is a radial basis function (RBF) SVM. It has two hyperparameters (C , γ). Cross-validation was used to tune the hyperparameters. Given a hyperparameter space $C : [1, 30]$ and $\gamma : [0.1, 1.0E-5]$, a pair of parameters is selected from the hyperparameter space by the cross-validation in each training and validation iteration. The optimal parameters selected for SVM in this article are $C : 4.8$ and $\gamma : 10E-4.5$.

The hyperparameter of an HMM is the number of hidden components. We searched for the number of hidden components from 0 to 20. It was found that 5 is the optimal number of hidden components.

DTW usually works with kNN to perform classification. The implementation in this article combined kNN and DTW by replacing the Euclidean distance measurement with the DTW distance measurement. The hyperparameter of kNN was the number of neighbors k and that of DTW was the *max warping window*. After tuning with cross-validation, the optimal hyperparameters are $k : 1$ and *max warping window*: 18.

As shown in Table II, both the LSTM network and the TCN are superior to these three algorithms in all three metrics. The TCN achieved the best results: 99.49% in OA, 99.53% in Recall, and 99.51% in F1.

C. Comparison With Related Work

Ma *et al.* [42] also used a 2-D LIDAR to perform HAR. They developed a seq2seq model to perform the classification of 17 activities and achieved 88% overall accuracy. We cannot perform a completely direct comparison between the work in [42] and ours. The layout of the kitchen and the LIDAR used in [42] are different from ours. The structure of the seq2seq model implemented in [42] is unclear as the way of trajectory collection and the number of their trajectory samples. However, we still can argue that our approach is able to perform multiple people tracking and it achieved very good accuracy in nearly real-time HAR. The work in [42] did not consider the interference from nonhuman objects and it can only be used to recognize the activities of a single person. A trajectory can be generated by several activities. In [42], it is not clear how the authors decided the number of activities that the seq2seq model output. It is more suitable to represent each activity by using a small trajectory segment, which also can improve real-time capability in activity recognition.

D. Analysis

From the above evaluations, it can be seen that deep learning has great performance in trajectory-based activity

classifications in comparison with the traditional algorithms. Trajectory augmentation is an effective way to improve the classification accuracy when the number of trajectories is small and imbalanced. The TCN performed better than the LSTM in trajectory classification. Because the length of the trajectory samples is only 25 (point), the capability of TCNs in learning long-range temporal patterns was not fully released. The high accuracy achieved in this article infers that indoor trajectory-based HAR is applicable. Of course, it also has a relationship with the layout of indoor scenarios. For a more complex indoor layout, the performance of activity recognition probably will decline.

VI. CONCLUSION

In this article, we proposed to perform HAR by using a 2-D LIDAR. We used the DBSCAN to cluster the LIDAR points and perform human and nonhuman classification-based upon 15 geometric features of the clusters. We further applied the Kalman filter to perform multiple people tracking and obtain their trajectories. We used a sliding window to perform trajectory segmentation. For overcoming the problem of unbalanced classes, we proposed trajectory augmentation by using spatial transformation and adding Gaussian noise. Finally, we built two networks (an LSTM and a TCN) to perform HAR upon the trajectory samples. Our proposed TCN achieved a very good result of 99.49% overall accuracy, which is slightly superior to the LSTM and much higher than the results achieved by SVM, HMM, and DTW. We compared this article to the state-of-the-art approach for HAR using LIDAR. Our proposed networks outperform the state-of-the-art approach.

With the coming age of the Internet of Things, various electronic sensors are being connected wirelessly. One future research direction is on fusing the data acquired from different sensors in order to achieve more efficient and accurate HAR [49]. In modern life, people switch between different scenarios, such as homes, offices, shops, etc. Therefore, sensor fusion, including wearable sensors (accelerometers, GPS receivers, gyroscopic sensors, etc.) and ambient sensors (RFID, WiFi, camera, etc.) in those scenarios is necessary to achieve ubiquitous HAR. Another research direction is on exploring unsupervised and semisupervised learning with a smaller amount of training data in order to make HAR more applicable in real scenarios. Finally, it is important to protect users' privacy while performing HAR. The current HAR systems need to upload the sensor data to servers or clouds, the sensor data could be illegally used resulting in an intrusion of users' privacy. One solution would be to deploy the HAR system on clients without uploading the sensor data and another one would be performing sensor data encryption before uploading without compromising performance.

In our future work, we will attempt to overcome the problem of unobstructed Line of Sight (LOS). For object detection and tracking, it is necessary to ensure there is no obstruction between the LIDAR and targets. With a single LIDAR, the moving targets are easily blocked by obstacles or other moving targets that obstruct their LOS. For example, in our

research, a person can block the LOS of another person frequently, this leads to a broken and uncontinuous trajectory. For overcoming this problem, we will use multiple LIDARs to perform human detection and tracking from a multiperspective. A trajectory-based HAR relies on trajectory pattern analysis. However, some microactivities, such as eating, drinking, calling, etc., cannot be recognized by only human trajectory. While wearable sensors or Radars are able to recognize them. So we will attempt to combine LIDARs with wearable sensors or Radars to perform both microactivity and macroactivity recognition simultaneously.

ACKNOWLEDGMENT

The authors would like to thank Dr. Kristou Mehrez and Hokuyo Automatic company for providing the LIDAR for their research.

REFERENCES

- [1] F. Luo, E. Bodanese, and S. Poslad, *Kitchen Activity Detection for Healthcare Using a Low-Power Radar-Enabled Sensor Network*, Inst. Elect. Electron. Eng., Piscataway, NJ, USA, 2019.
- [2] A. Rafii, S. B. Gokturk, C. Tomasi, and F. Sürücü, "Gesture recognition system using depth perceptive sensors," U.S. Patent Appl. 10242 255, Mar. 26, 2019.
- [3] S. Wang, J. Song, J. Lien, I. Poupyrev, and O. Hilliges, "Interacting with soli: Exploring fine-grained dynamic gesture recognition in the radio-frequency spectrum," in *Proc. ACM 29th Annu. Symp. User Interface Softw. Technol.*, 2016, pp. 851–860.
- [4] X. Song, X. Shao, R. Shibasaki, H. Zhao, J. Cui, and H. Zha, "A novel laser-based system: Fully online detection of abnormal activity via an unsupervised method," in *Proc. IEEE Int. Conf. Robot. Autom.*, 2011, pp. 1317–1322.
- [5] P. Vallabh and R. Malekian, "Fall detection monitoring systems: A comprehensive review," *J. Ambient Intell. Humanized Comput.*, vol. 9, no. 6, pp. 1809–1833, 2018.
- [6] J. C. Nunez, R. Cabido, J. J. Pantrigo, A. S. Montemayor, and J. F. Velez, "Convolutional neural networks and long short-term memory for skeleton-based human activity and hand gesture recognition," *Pattern Recognit.*, vol. 76, pp. 80–94, Apr. 2018.
- [7] M. S. Ibrahim, S. Muralidharan, Z. Deng, A. Vahdat, and G. Mori, "A hierarchical deep temporal model for group activity recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2016, pp. 1971–1980.
- [8] P.-G. Jung, G. Lim, S. Kim, and K. Kong, "A wearable gesture recognition device for detecting muscular activities based on air-pressure sensors," *IEEE Trans. Ind. Informat.*, vol. 11, no. 2, pp. 485–494, Apr. 2015.
- [9] J. Qi, P. Yang, M. Hanneghan, S. Tang, and B. Zhou, "A hybrid hierarchical framework for gym physical activity recognition and measurement using wearable sensors," *IEEE Internet Things J.*, vol. 6, no. 2, pp. 1384–1393, Apr. 2019.
- [10] P. C. Shih, K. Han, E. S. Poole, M. B. Rosson, and J. M. Carroll, "Use and adoption challenges of wearable activity trackers," in *Proc. IConf.*, 2015, pp. 6–8.
- [11] J. Huberty, D. K. Ehlers, J. Kurka, B. Ainsworth, and M. Buman, "Feasibility of three wearable sensors for 24 hour monitoring in middle-aged women," *BMC Women's Health*, vol. 15, no. 1, p. 55, 2015.
- [12] Z. Ma, B. Wu, and S. Poslad, "A WiFi RSSI ranking fingerprint positioning system and its application to indoor activities of daily living recognition," *Int. J. Distrib. Sensor Netw.*, vol. 15, no. 4, p. 14, 2019.
- [13] F. Luo, S. Poslad, and E. Bodanese, "Human activity detection and coarse localization outdoors using micro-doppler signatures," *IEEE Sens. J.*, vol. 19, no. 18, pp. 8079–8094, Sep. 2019.
- [14] H. Wang, D. Zhang, Y. Wang, J. Ma, Y. Wang, and S. Li, "RT-Fall: A real-time and contactless fall detection system with commodity WiFi devices," *IEEE Trans. Mobile Comput.*, vol. 16, no. 2, pp. 511–526, Feb. 2017.
- [15] L. Ren, Y. S. Koo, H. Wang, Y. Wang, Q. Liu, and A. E. Fathy, "Noncontact multiple heartbeats detection and subject localization using UWB impulse doppler radar," *IEEE Microw. Wireless Compon. Lett.*, vol. 25, no. 10, pp. 690–692, Oct. 2015.

- [16] E. Aksan and O. Hilliges, "STCN: Stochastic temporal convolutional networks," 2019. [Online]. Available: arXiv:1902.06568.
- [17] S. C. Mukhopadhyay, "Wearable sensors for human activity monitoring: A review," *IEEE Sens. J.*, vol. 15, no. 3, pp. 1321–1330, Mar. 2015.
- [18] M. Buettner, R. Prasad, M. Philipose, and D. Wetherall, "Recognizing daily activities with RFID-based sensors," in *Proc. ACM 11th Int. Conf. Ubiquitous Comput.*, 2009, pp. 51–60.
- [19] M. Sridharan, J. Bigham, P. M. Campbell, C. Phillips, and E. Bodanese, "Inferring micro-activities through wearable sensing for ADL recognition of home-care patients," *IEEE J. Biomed. Health Informat.*, vol. 24, no. 3, pp. 747–759, Mar. 2020.
- [20] W. Xi *et al.*, "Device-free human activity recognition using CSI," in *Proc. ACM 1st Workshop Context Sens. Activity Recognit.*, 2015, pp. 31–36.
- [21] S. Sempena, N. U. Maulidevi, and P. R. Aryan, "Human action recognition using dynamic time warping," in *Proc. IEEE Int. Conf. Elect. Eng. Informat.*, 2011, pp. 1–5.
- [22] E. J. Keogh and C. A. Ratanamahatana, "Exact indexing of dynamic time warping," *Knowl. Inf. Syst.*, vol. 7, no. 3, pp. 358–386, 2005.
- [23] S. Kamal, A. Jalal, and D. Kim, "Depth images-based human detection, tracking and activity recognition using spatiotemporal features and modified HMM," *J. Elect. Eng. Technol.*, vol. 11, no. 3, pp. 1921–1926, 2016.
- [24] W. Wang, A. X. Liu, M. Shahzad, K. Ling, and S. Lu, "Understanding and modeling of WiFi signal based human activity recognition," in *Proc. 21st Annu. Int. Conf. Mobile Comput. Netw.*, 2015, pp. 65–76.
- [25] C. A. Ronao and S.-B. Cho, "Recognizing human activities from smartphone sensors using hierarchical continuous hidden Markov models," *Int. J. Distrib. Sensor Netw.*, vol. 13, no. 1, pp. 1–16, 2017.
- [26] Z. Chen, Q. Zhu, Y. C. Soh, and L. Zhang, "Robust human activity recognition using smartphone sensors via CT-PCA and online SVM," *IEEE Trans. Ind. Informat.*, vol. 13, no. 6, pp. 3070–3080, Dec. 2017.
- [27] P. Paul and T. George, "An effective approach for human activity recognition on smartphone," in *Proc. IEEE Int. Conf. Eng. Technol. (ICETECH)*, 2015, pp. 1–3.
- [28] S. Balli, E. A. Sağbaş, and M. Peker, "Human activity recognition from smart watch sensor data using a hybrid of principal component analysis and random forest algorithm," *Meas. Control*, vol. 52, nos. 1–2, pp. 37–45, 2019.
- [29] A. Kamel, B. Sheng, P. Yang, P. Li, R. Shen, and D. D. Feng, "Deep convolutional neural networks for human action recognition using depth maps and postures," *IEEE Trans. Syst., Man, Cybern., Syst.*, vol. 49, no. 9, pp. 1806–1819, Sep. 2019.
- [30] J. Tang, X. Shu, R. Yan, and L. Zhang, "Coherence constrained graph LSTM for group activity recognition," *IEEE Trans. Pattern Anal. Mach. Intell.*, early access, Jul. 15, 2019, doi: 10.1109/TPAMI.2019.2928540.
- [31] Y. Guo, D. Tao, W. Liu, and J. Cheng, "Multiview cauchy estimator feature embedding for depth and inertial sensor-based human action recognition," *IEEE Trans. Syst., Man, Cybern., Syst.*, vol. 47, no. 4, pp. 617–627, Apr. 2017.
- [32] C. Lea, M. D. Flynn, R. Vidal, A. Reiter, and G. D. Hager, "Temporal convolutional networks for action segmentation and detection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2017, pp. 156–165.
- [33] K. L. Y. Siang and S. W. Khor, "Path clustering using dynamic time warping technique," in *Proc. IEEE 8th Int. Conf. Comput. Technol. Inf. Manag. (NCM ICNIT)*, vol. 1, 2012, pp. 449–452.
- [34] J. Liang, L. Jiang, J. C. Niebles, A. Hauptmann, and L. Fei-Fei, "Peeking into the future: Predicting future person activities and locations in videos," 2019. [Online]. Available: arXiv:1902.03748.
- [35] Y. Wang, D. Wang, and F. Chen, "Abnormal behavior detection using trajectory analysis in camera sensor networks," *Int. J. Distrib. Sensor Netw.*, vol. 10, no. 1, 2013, Art. no. 839045.
- [36] Q. Gao and S. Sun, "Trajectory-based human activity recognition with hierarchical Dirichlet process hidden Markov models," in *Proc. IEEE China Summit Int. Conf. Signal Inf. Process.*, 2013, pp. 456–460.
- [37] T. Fuse and K. Kamiya, "Statistical anomaly detection in human dynamics monitoring using a hierarchical Dirichlet process hidden Markov model," *IEEE Trans. Intell. Transp. Syst.*, vol. 18, no. 11, pp. 3083–3092, Nov. 2017.
- [38] L. Huang, Q. Li, and Y. Yue, "Activity identification from GPS trajectories using spatial temporal POIs' attractiveness," in *Proc. 2nd ACM SIGSPATIAL Int. Workshop Location Soc. Netw.*, 2010, pp. 27–30.
- [39] W. Huang, M. Li, W. Hu, G. Song, X. Xing, and K. Xie, "Cost sensitive GPS-based activity recognition," in *Proc. 10th Int. Conf. Fuzzy Syst. Knowl. Disc. (FSKD)*, 2013, pp. 962–966.
- [40] H. Martin, D. Bucher, E. Suel, P. Zhao, F. Perez-Cruz, and M. Raubal, "Graph convolutional neural networks for human activity purpose imputation from GPS-based trajectory data," in *Proc. 32nd Conf. Neural Inf. Process. Syst. (NIPS)*, 2018, pp. 1–6.
- [41] A. Panangadan, M. J. Mataric, and G. S. Sukhatme, "Tracking and modeling of human activity using laser rangefinders," *Int. J. Soc. Robot.*, vol. 2, no. 1, pp. 95–107, 2010.
- [42] Z. Ma, J. Bigham, S. Poslad, B. Wu, X. Zhang, and E. Bodanese, "Device-free, activity during daily life, recognition using a low-cost LIDAR," in *Proc. IEEE Global Commun. Conf. (GLOBECOM)*, 2018, pp. 1–6.
- [43] P. H. Ahmad and S. Dang, "Performance evaluation of clustering algorithm using different datasets," *Int. J. Adv. Res. Comput. Sci. Manag. Stud.*, vol. 3, no. 1, pp. 167–173, 2015.
- [44] A. Leigh, J. Pineau, N. A. Olmedo, and H. Zhang, "Person tracking and following with 2D laser scanners," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, 2015, pp. 726–733.
- [45] R. J. Meinhold and N. D. Singpurwalla, "Understanding the Kalman filter," *Amer. Stat.*, vol. 37, no. 2, pp. 123–127, 1983.
- [46] S. Bai, J. Z. Kolter, and V. Koltun, "An empirical evaluation of generic convolutional and recurrent networks for sequence modeling," 2018. [Online]. Available: arXiv:1803.01271.
- [47] *Scanning Laser Range Finder Smart-URG Mini UST-10Lx (UUST003) Specification*, Hokuyo, Osaka, Japan, 2015.
- [48] S. Yadav and S. Shukla, "Analysis of k -fold cross-validation over hold-out validation on colossal datasets for quality classification," in *Proc. IEEE 6th Int. Conf. Adv. Comput. (IACC)*, 2016, pp. 78–83.
- [49] J. Qi, P. Yang, L. Newcombe, X. Peng, Y. Yang, and Z. Zhao, "An overview of data fusion techniques for Internet of Things enabled physical activity recognition and measure," *Inf. Fusion*, vol. 55, pp. 269–280, Mar. 2020.



Fei Luo received the B.Sc. degree in geographic information system from the Jiangxi University of Science and Technology, Ganzhou, China, the M.Sc. degree in surveying and mapping from Wuhan University, Wuhan, China, and the Ph.D. degree in electronic engineering from the Queen Mary University of London, London, U.K., in 2020.

He is currently a Postdoctoral Researcher with the School of Geospatial Engineering and Science, Queen Mary University of London. His research interests include geographic information systems, human activity detection, remote sensing, and machine learning.



Stefan Poslad received the B.Sc. degree in physics from the University of Southampton, Southampton, U.K., the M.Sc. degree in medical physics from the University of Aberdeen, Aberdeen, U.K., and the Ph.D. degree from Newcastle University, Newcastle upon Tyne, U.K.

Since 2000, he has been a Principal Investigator of 12 funded collaborative research projects, and has received funding from the U.K. Research Council, Europe, Norway, U.S., and from the industry.



Eliane Bodanese received the M.Sc. degree in electrical engineering from the Federal University of Santa Catarina, Florianópolis, Brazil, and the Ph.D. degree in electronic engineering from the Queen Mary University of London, London, U.K.

She is a Senior Lecturer with the School of Electronic Engineering and Computer Science, Queen Mary University of London, London, U.K. Her research interests include human activity detection and recognition through minimal nonintrusive sensing, indoor and outdoor localization, and communication support for emergencies.

Dr. Bodanese is a member of the Communications System Research Group.