

Research Article

Blind Travel Prediction Based on Obstacle Avoidance in Indoor Scene

Zhiqiang Lv , Jianbo Li , Haoran Li , Zhihao Xu , and Yue Wang 

College of Computer Science & Technology, Qingdao University, 266071, China

Correspondence should be addressed to Jianbo Li; lijianbo@qdu.edu.cn

Received 17 February 2021; Accepted 6 June 2021; Published 23 June 2021

Academic Editor: Xiao Zhang

Copyright © 2021 Zhiqiang Lv et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Blind people have intelligent tools to rely on for travel with the development of navigation technology. The GPS navigation, blind track, etc., are tools that blind people often use when traveling outdoors. However, indoor navigation tools and technology for blind people are lacking. We propose an obstacle avoidance algorithm and a spatial-temporal model of trajectory prediction for the indoor travel task of the blind. The focus of this work is that it enables the blind to accurately avoid obstacles and achieve high accuracy trajectory prediction aiming at the unique movement characteristics of the blind. We set up a variety of baselines to conduct an experimental evaluation on a dataset of blind trajectories in a multistorey shopping mall. The experimental results show the advantages of the data model and predictive model of this work.

1. Introduction

With the rapid development of the transportation and automobile industry, the trajectory prediction has become a focus in the field of transportation big data. It helps people plan better travel routes as well as saving more manpower and material resources. More importantly, the public basic resources are allocated more precisely with the help of trajectory prediction. Blind people live in a dark world, which brings great difficulties to work, life, and social activities [1]. How to walk safely is the biggest problem in the life of the blind. At present, there are many researchers in the world who solve the problem of outdoor navigation for the blind, such as voice broadcast systems in public places (blind roads and bus stops) and GPS (Global Positioning System) for outdoor navigation for the blind. However, there are few results in indoor navigation for the blind. The GPS navigation functions are already very mature. However, GPS is mainly used for outdoor travel activities for blind people. Even if they go out, they usually go to hospitals, hotels, and other indoor buildings. So, the indoor scenes are the main activity areas for blind people and they have their own different channels and obstacle spatial distribution. Moving indoors without GPS is a problem for blind people. Therefore, the research of indoor navigation projects has

extremely important social significance and research value. There are two important considerations in indoor navigation for the blind, namely, the accurate prediction of the trajectory of the blind and the accurate avoidance of indoor static objects.

Blind people need auxiliary equipment to issue motion instructions for them during walking, so the trajectory prediction for blind people is an important basis for generating motion instructions. Pedestrian trajectory prediction models are mainly divided into traditional mathematical-statistical models and data-driven neural network models [2]. Traditional mathematical-statistical models rely on artificially designed features to model pedestrian actions and goals. The Social Force Model (SFM) proposed by Helbing and Molnar [3] transforms the interaction between pedestrians and pedestrian goals into gravitation and repulsion. This work believes that the goal of pedestrians can attract pedestrians to the goal. The repulsion among pedestrians prevents pedestrian collision. Trautman and Krause [4] improve the SFM with an interactive Gaussian process. They use the Gaussian process to predict the trajectory of each pedestrian and calculate the probability of the prediction result according to the potential function of SFM. The Markov model can make probabilistic spatial-temporal prediction of pedestrian

trajectory [5] [6]. The training process of the model can dynamically adjust the training parameters with the help of reinforcement learning [7]. It can make the prediction process consider the physical influence of the outside world and make the predicted trajectory closer to the actual trajectory. The above methods have the advantages of simplicity, intuitiveness, and low complexity, but their process of building a model is too sensitive to calculate parameters, and the generalization ability of the model is weak. More importantly, the above methods can only simulate the short-term reaction of pedestrians and cannot consider the long-term historical information of the location.

In data-driven forecasting tasks, the recurrent neural network (RNN) has obvious advantages over traditional mathematical-statistical models [8], especially in the long-term and time-dependent feature calculation process. The RNN is a neural network used to process sequence data. Compared with the general neural network, it can process the data of the sequence change. The Long Short-Term Memory (LSTM) [9] is a special RNN that can solve the problems of gradient disappearance and gradient explosion in the training process of long sequences. More importantly, the LSTM can perform better in longer sequences than ordinary RNNs. The LSTM can not only realize the sequence prediction of pedestrian position but also calculate the mutual influence among different pedestrians [10]. However, the LSTM has the disadvantage that the RNN cannot capture the high-level spatial-temporal structure [11]. In order to overcome this disadvantage and maintain the characteristics of pedestrian trajectory, Alahi et al. [12] propose the Social Long Short-Term Memory (S-LSTM) model. The S-LSTM collects the hidden state of adjacent pedestrians by introducing a social pooling layer and shares the hidden information of adjacent pedestrians by the spatial distance of a grid. In order to reduce information loss, Vemula et al. [13] replaced the social pooling layer with the social attention layer. The social attention layer forms interactive features by assigning weights among pedestrians in the grid. Unlike the walking process of normal people, the walking speed of blind people is very slow, because they can make the next decision to walk only after they have fully explored the current road. The above-mentioned research works all focus on predicting the trajectory of normal people, so they lack feature calculations for the movement characteristics of blind people.

The trajectory prediction of a blind person realizes the prediction of the short-term future position of the blind based on the characteristics of motor behavior. Any building should be regarded as an obstacle for the blind. Blind people should avoid obstacles fully when walking [14]. The above movement characteristics of the blind determine the focus of this work on how to achieve accurate prediction of the position of the blind and how to avoid obstacles. The movement characteristics of the blind determine the focus of this work on how to realize the accurate prediction of the position of the blind while avoiding obstacles. Wang et al. [15] propose a path planning algorithm for blind navigation systems. It uses the Dijkstra algorithm [16] as the basic algorithm and relational database as the storage mode. The algorithm uses a

multifactor fuzzy algorithm to calculate the weight of obstacles in the road network. Its core is to make an adjacency matrix according to the spatial distribution of obstacles in the road network. Finally, it makes the topological structure diagram of the obstacle network. However, this obstacle network is a method of representing the position of obstacles in a local area. When the position of the blind person changes, the obstacle network needs to be recalculated. This design mode has high computational complexity and cannot consider the spatial distribution of obstacles on a global scale. The Graph Convolutional Network (GCN) [17] shows superiority in the representation of global spatial relations. The theory of convolutional neural network (CNN) [18] is to use random and shared convolution kernel to get the weighted sum of pixels. Then, it uses backpropagation to optimize the convolution kernel parameters to automatically extract features. However, many data in real life are stored in the form of graphs, such as social network information, knowledge graphs, protein networks, and the World Wide Web. The form of these graph networks is not like the images that are neatly arranged in matrix form but unstructured data. The GCN has a general paradigm for calculating graph features. More importantly, the GCN can use the adjacency matrix representing the connectivity of nodes to represent the connectivity among the entire spatial location [19]. This model can be used to calculate the spatial distribution of obstacles in indoor spaces. For the blind, the blind should avoid obstacles, which means that the location of the obstacle does not have connectivity.

In response to the above research, we propose a Blind Trajectory Prediction Model (BlindTPM). The BlindTPM is used for the tasks of blind trajectory prediction and obstacle avoidance in indoor scenes. The main contributions of this work are as follows:

- (i) We propose a spatial-temporal model for pedestrian trajectory prediction. The spatial convolution block captures the spatial relative characteristics of roads and obstacles. Different from the existing pedestrian trajectory model, this model combines the unique position change characteristics of the blind to realize the trajectory prediction task. According to the slow motion of blind people, the temporal convolution block designed by this model can capture long-term historical information of pedestrian trajectories
- (ii) We propose a trajectory algorithm for obstacle avoidance. The algorithm calculates the spatial distribution and connectivity of obstacles with the help of the GCN and connectivity adjacency matrix
- (iii) We design three data-driven methods to verify the performance of BlindTPM, including a trajectory prediction method based on abscissa and ordinate, a trajectory prediction method based on a grid map, and a trajectory prediction method based on a grid map with obstacle distribution. We model the trajectory data of blind people in a multifloor shopping mall. Compared with the existing

baselines, the BlindTPM has obvious performance advantages

2. Data Design

We design three data-driven methods to complete the task of blind trajectory prediction, including a trajectory prediction method based on abscissa and ordinate, a trajectory prediction method based on a grid map, and a trajectory prediction method based on a grid map with obstacle distribution.

2.1. Abscissa and Ordinate. The abscissa and ordinate are used to represent the position of the blind in the design idea of this method, as shown in Figure 1. The $X = \{x_1, x_2, x_3, \dots, x_i\}$ and $Y = \{y_1, y_2, y_3, \dots, y_i\}$. Because this work is aimed at a fixed blind moving scene indoors, the X and Y are relative positions to a fixed origin, not latitude and longitude. The t is the time step of the blind position change. This method implements the most basic trajectory prediction task, which uses t consecutive positions to predict $t + n$ positions in the future, as shown in

$$(X', Y') = \sigma'(\text{BPT}(\sigma(X, Y))). \quad (1)$$

The $X' = \{x_{t+1}, x_{t+2}, \dots, x_{t+n}\}$ and $Y' = \{y_{t+1}, y_{t+2}, \dots, y_{t+n}\}$. The BPT is the modeling method in this work. The σ and σ' represent the data standardization and destandardization process, respectively. Different evaluation indicators usually have different dimensions and units in the field of machine learning. In order to eliminate the dimensional influence between indicators, we need to standardize the data so that the data indicators can be compared. Because the abscissa and ordinate data representing the position of the blind are floating-point numbers and the data changes are limited to a small range, so we use the Z -score as the method of data standardization, as shown in Formula (2). The μ represents the mean of the overall data, and the δ represents the standard deviation of the overall data. The calculation process of Z -score is simple, and it eliminates the impact of data magnitude.

$$Z = \frac{(X | Y) - \mu}{\delta}. \quad (2)$$

2.2. Grid Map. There is a very obvious difference between the walking trajectory characteristics of a normal person and a blind person. Figure 2 shows the changes of abscissa and ordinate of two trajectories in the same scene according to time steps. The time step interval of the two trajectories is the same, which is two seconds. The trajectory of a normal person is basically a linear change, which has obvious characteristics of change. However, the position of the blind does not change frequently due to the constant movement and the need to explore the road. This situation is reflected by the continuous same value of the abscissa and ordinate of the trajectory in Figure 2. Finally, it causes a loss of accuracy due to a large number of sample repetitions in the weight calculation process of weight of actual trajectory.

In order to deal with the above situation, we use the local sampling [20] method to clean the dataset of the blind trajectory, as shown in Figure 3. We use sample points to replace multiple locally unchanged points in the abscissa and ordinate. The data shows obvious trajectory features after being cleaned, as shown in Figure 4(a). We can know that the position of a blind person can be represented by the label of a grid when the indoor scene is mapped to a grid. This method transforms the regression prediction task of this work into a classification task when the indoor scene of the blind movement is relatively single and the number of divided grids is large.

The calculation method of dividing the track points into grid labels is shown in Formula (3). The α is the coordinate dimension represented by a grid label, which is equal to 2 in this work. Formula (3) is applicable to the grid label calculation process of abscissa and ordinate. More importantly, it takes into account the positive and negative values of coordinate data.

$$l_i^{(x)} = \begin{cases} \left\lceil \frac{\max(X) - x_i + 1}{\alpha} \right\rceil, & x_i > 0, \\ \left\lceil \frac{|\min(X)| + x_i + 1}{\alpha} \right\rceil, & x_i < 0. \end{cases} \quad (3)$$

According to the above method, we can get the grid labels of abscissa ($l_i^{(x)}$) and ordinate ($l_i^{(y)}$). We expect to obtain n positions of the blind in the future according to the historical trajectory, so the $L^{(x)}$ and $L^{(y)}$ in Formula (4) are sequence data. The $L^{(x)} = \{l_{i+1}^{(x)}, l_{i+2}^{(x)}, \dots, l_{i+n}^{(x)}\}$ and $L^{(y)} = \{l_{i+1}^{(y)}, l_{i+2}^{(y)}, \dots, l_{i+n}^{(y)}\}$. Generally speaking, a location point is described by its abscissa and ordinate in the 2-dimensional plane. So, we take this into consideration when evaluating the accuracy of the model. Only when the predicted values of the grid labels of the abscissa and ordinate are the same as the true values in the same time step can it be considered as a successful prediction.

$$(L^{(x)}, L^{(y)}) = \text{LogSoftmax}(\text{BPT}(X, Y)). \quad (4)$$

2.3. Obstacle Distribution. The spatial distribution of obstacles is an important factor affecting the walking process of blind people indoors. Figure 5(a) shows the spatial distribution of obstacles (buildings, decorations, etc.) in the actual indoor scene. Among them, the black areas represent obstacles. Figure 5(b) is a standardized obstacle distribution map, which is derived from the data cleaning of (a). We must admit that (b) there is a certain information error, so we try our best to increase the number of obstacles to avoid the problem of missing obstacles. Finally, the grid area where obstacles exist is assigned the value 0 and the other grid areas are assigned the value 1, as shown in Figure 5(c).

We design an obstacle avoidance algorithm for blind people based on the connectivity among grids, as shown in Figure 6. The main theory of the algorithm is to use the connectivity among grids to construct an adjacency matrix to

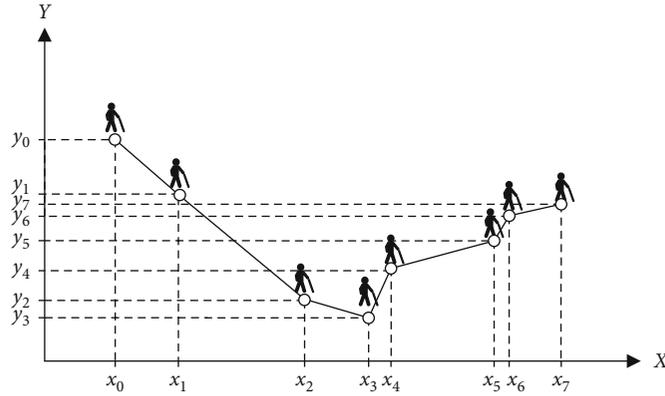


FIGURE 1: Basic track position representation. The location point changes according to time. The X and Y represent the abscissa and ordinate of the position, respectively.

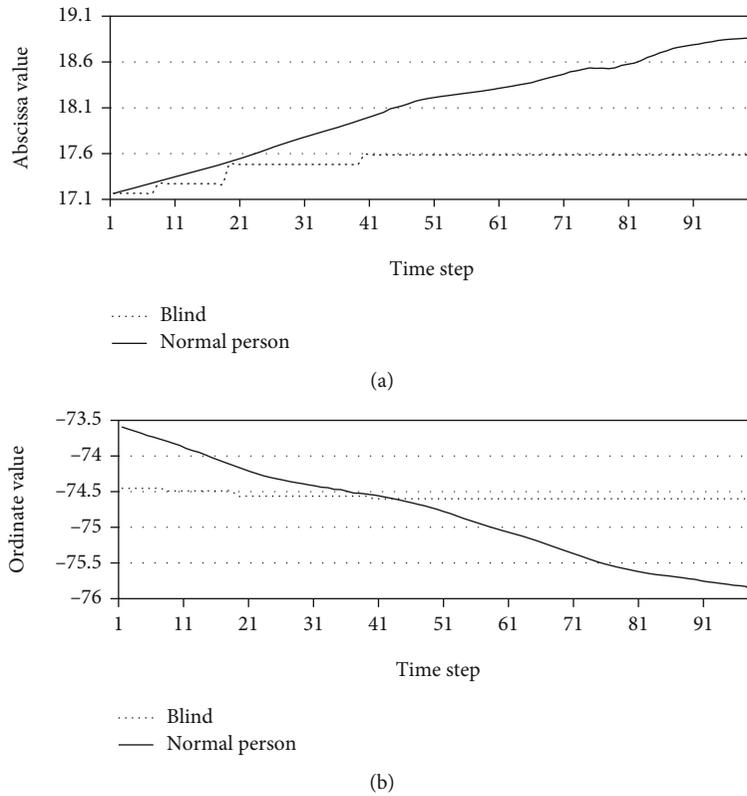


FIGURE 2: The difference between the trajectory characteristics of the blind and normal person: (a) the change curve of the abscissa of trajectory with time and (b) the change curve of the ordinate of trajectory with time.

capture the global spatial distribution characteristics of roads and obstacles in indoor scenes. The adjacency matrix is a data structure used to describe the relationship between vertices and edges. It is essentially a two-dimensional array and is suitable for dealing with the relationship among the smallest data units. Blind people can walk in up to nine directions in a grid, including front, back, left, right, front left, front right, back left, back right, and motionless. These nine directions correspond to the eight adjacent grids of a grid and the grid where it is located. However, the presence of obstacles (the black grid in Figure 6(a)) makes the connectivity of a grid to nine directions uncertain. According

to the obstacle spatial distribution grid in Figure 5(c) and grid connectivity rule, we design an algorithm for calculating grid connectivity, which is suitable for the computing process of the connectivity adjacency matrix among grids with obstacles, as shown in Algorithm 1. The output A of Algorithm 1 is a symmetric adjacency matrix, as shown in Figure 6(b). Its symmetry is shown in that the connectivity between one grid and another grid is equivalent in the forward and reverse interaction process. If there are obstacles in a grid, the grid is defined as that it cannot interact with any other grids and 9 directions are assigned the value 0.

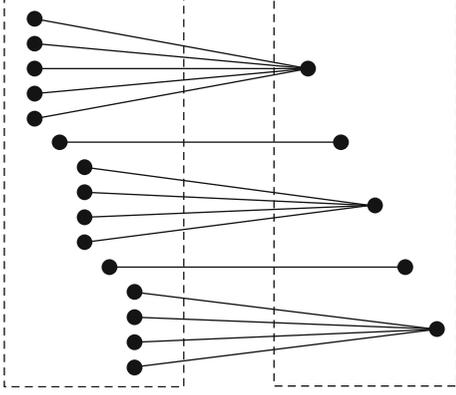


FIGURE 3: Sample trajectory design. The left side is the original trajectory, and the right side is the sample trajectory.

3. Model Design

We design a deep spatial-temporal model (BlindTPM) that can train, evaluate, and predict the trajectory data of blind people. The BlindTPM consists of three blocks, including spatial convolution block, temporal convolution block, and estimation block, as shown in Figure 7. The spatial convolution block is mainly used to calculate the spatial distribution of trajectories and obstacles. The temporal convolution block is mainly used to calculate the time recursive characteristics of trajectory data. The estimation block is mainly used to reduce the global error and local error of the trajectory prediction result.

3.1. Spatial Convolution Block. The walking process of the blind is complex and diverse. Although the prediction of straight lines is simple, it is often the case that the traveling route becomes a complex curve due to the influence of turns and obstacle distribution. This makes it particularly important to capture the spatial relationship of positions in the feature extraction process of trajectory prediction. Therefore, it is necessary to perform another round of extraction of feature from the trajectory and strengthen the weight of the local curve of the trajectory before making temporal predictions, instead of using standardized position data directly [21]. First of all, the data of abscissa and ordinate are fused to calculate the feature of spatial dependencies, as shown in

$$p_i = \sigma(\text{cat}(\delta^{x_k} \times x_i + \mu^{x_k}, \delta^{y_k} \times y_i + \mu^{y_k})). \quad (5)$$

The abscissa and ordinate of the i th point are calculated with the standard deviation δ and mean μ of the k th trajectory sequence, which represents standardization of position data. The result of cat that fuses the abscissa and ordinate into the same dimension is closer to the real result compared to the result of processing abscissa or ordinate matrix separately. The correlation between the spatial dependence strength of the nodes of the blind trajectory and the contextual information of the trajectory feature is very important. Traditional convolutional networks usually use downsampling to perform high-dimensional feature calculations. However, the fixed receptive field of the downsampling

method will cause the loss of edge feature information. Particularly for the trajectory of the blind, the walking process of the blind is very slow, which leads to a lot of trajectory nodes in a certain period of time. The above process makes the weight of the trajectory involve more parameters, and the parameter information becomes very easy to lose. Therefore, we build multilayers of dilated convolution to achieve the process of capturing the complex spatial dependence of trajectory of the blind, as shown in Figure 8. Filter kernels are depicted in a square with a grid pattern in each layer. The black cells in dilated kernel represent valid weights. The dilation factor of the three-layer dilation convolutional network is changed by 2^2 , 2^1 , and 2^0 . The biggest advantage of a dilation convolutional network is that it can expand the receptive field exponentially without losing feature information. The $\{d\}$ is the distance features of the first and the nonfirst nodes of every trajectory.

We combine the o^{conv} and the data which are the distance of the i th node and the first node of the k th trajectory after the activation of the σ (ReLU function) in order to reduce the mutual dependence of parameters and alleviate the problem of gradient disappearance due to overfitting, as shown in

$$O = \sigma\left(\text{cat}\left(o^{\text{conv}}, \delta^{d_k} \times d_i + \mu^{d_k}\right)\right). \quad (6)$$

3.2. Obstacle Grid Spatial Distribution. In order to add the obstacle grid to the modeling process, we use the GCN to fuse the connectivity adjacency matrix with the feature data. The essence of the process of the GCN is that each node of graph is changing its state all the time due to the influence of neighbors and further nodes. The more closely related other nodes have a greater impact on the original node. The method of Laplace can transfer the strength of features in the GCN in proportion to the state difference among them. In order to add the influence of the original node on itself into the calculation process, we use an improved version of the method of Laplace, as shown in

$$\text{La} = O^{-1/2} A O^{-1/2}. \quad (7)$$

The A represents a connectivity adjacency matrix with a self-connected state. We take the output of the spatial convolution block as the degree distribution of the nodes (O). The above formula introduces its own degree matrix to solve the problem of self-transmission and realizes the normalization operation of the adjacency matrix by multiplying the two sides of the adjacency matrix by the degree root of the node and taking the inverse. The original spectrogram convolution implements the filter of the product of each node and the Fourier transform. However, the eigenvectors are high order and the eigendecomposition of the Laplacian matrix is very inefficient in the decomposition process of large graph structures [22]. So, we use the K -order Chebyshev polynomials to approximate the optimization of the Laplacian matrix, as shown in Formula (8). The calculation process of $T_i(\tilde{\text{La}})$ is shown in Formula (9), which represents the recursive definition of the Chebyshev polynomial. This method is called the K -localized [23] convolution algorithm, which ensures that


```

Input: The set of obstacle spatial distribution,  $L$ ; the side length of grid,  $N$ .
Output: The adjacency matrix of grid connectivity,  $A$ .
1   Initialize a matrix  $A$  with 0, the shape of  $A$  is  $(N^2 \times N^2)$ .
2   for each  $i \in [0, N^2 - 1]$  do
3     for each  $j \in [i, N^2 - 1]$  do
4       // Determine whether the current grid is an obstacle grid
5       if  $L_i == 0$  then  $A_{ij} = 0$ 
6       else
7         if  $i == j$  then  $A_{ij} = 1$ 
8         else if  $L_j == 1$  then
9           // Determine the connectivity of front.
10          if  $j == i - N$  then  $A_{ij}, A_{ji} = 1, 1$ 
11          // Determine the connectivity of left.
12          else if  $j == i - 1$  &  $i \% N != 0$  then  $A_{ij}, A_{ji} = 1, 1$ 
13          else if  $j == i + 1$  then
14            if  $i < N - 1$  then  $A_{ij}, A_{ji} = 1, 1$ 
15            end if
16            // Determine the connectivity of right.
17            if  $i \% N != 2$  &  $i > N - 1$  then  $A_{ij}, A_{ji} = 1, 1$ 
18            end if
19            // Determine the connectivity of front left.
20            else if  $j == i - N - 1$  &  $i \% N != 0$  then  $A_{ij}, A_{ji} = 1, 1$ 
21            // Determine the connectivity of front right.
22            else if  $j == i - N + 1$  &  $i \% N != 2$  &  $i > N - 1$  then  $A_{ij}, A_{ji} = 1, 1$ 
23            // Determine the connectivity of back.
24            else if  $j == i + N$  then  $A_{ij}, A_{ji} = 1, 1$ 
25            // Determine the connectivity of back left.
26            else if  $j == i + N - 1$  &  $i \% N != 0$  then  $A_{ij}, A_{ji} = 1, 1$ 
27            // Determine the connectivity of back right.
28            else if  $j == i + N + 1$  &  $i \% N != 2$  &  $i > N - 1$  then  $A_{ij}, A_{ji} = 1, 1$ 
29            end if
30          end if
31        end if
32      end for
33    end for
34  final
35  return  $A$ 

```

ALGORITHM 1: A computing algorithm of adjacency matrix of grid connectivity.

the current node only considers the impact of nodes in the K range on it and greatly reduces the time complexity.

$$g(\theta) \cdot o \approx \sum_{i=1}^k \theta_i \cdot T_i(\tilde{L}a) \cdot o, \quad (8)$$

$$T_i(\tilde{L}a) = \begin{cases} 2\tilde{L}a \cdot T_{i-1}(\tilde{L}a) - T_{i-2}(\tilde{L}a), \\ T_0(\tilde{L}a) = 1, \\ T_1(\tilde{L}a) = \tilde{L}a, \\ \tilde{L}a = \frac{2}{\lambda_{\max}} \times La - I_n. \end{cases} \quad (9)$$

3.3. Temporal Convolution Block. Recurrent neural networks have played a huge role in the field of time series data predic-

tion. However, traditional recurrent neural networks (RNN, LSTM, GRU, etc.) only involve a single-step calculation method. Single-step calculation has two disadvantages, namely, the complexity of the calculation process is very high and the performance of long-term prediction is low. The Temporal Convolution Network (TCN) [24] is designed based on the idea of convolutional neural network and parallelization. It overcomes the two inherent shortcomings of traditional recurrent neural networks. For blind person trajectory prediction, the TCN uses causal convolution to make all historical location points of the trajectory be associated with predicted future location points. The using of the dilated convolution enables the hidden layer to obtain a larger receptive field to establish the high-dimensional timing features of the blind trajectory. We build the temporal convolution block with 7 hidden layers to calculate the temporal features [21], as shown in Figure 9. Every hidden layer has one dilation factor and every factor is exponential growth

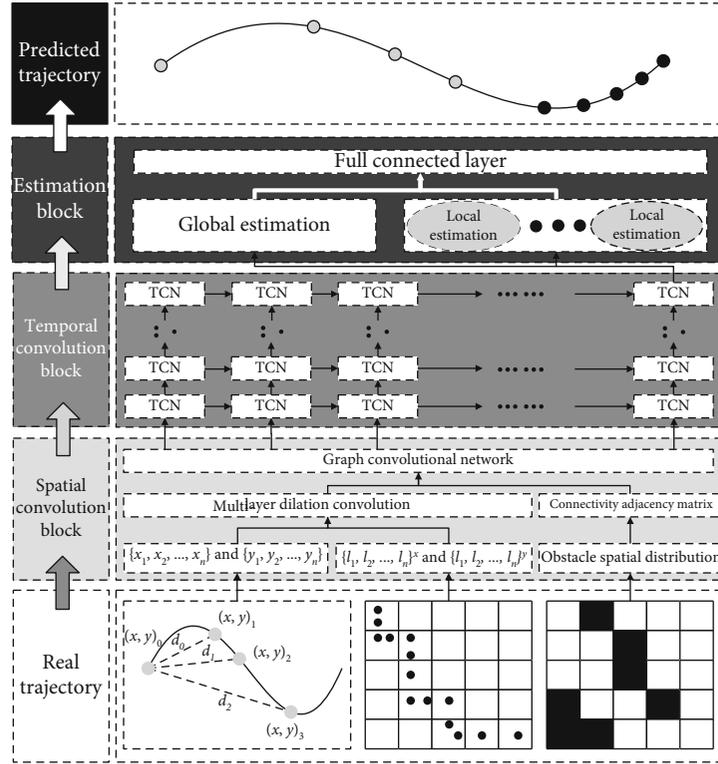


FIGURE 7: Model architecture design.

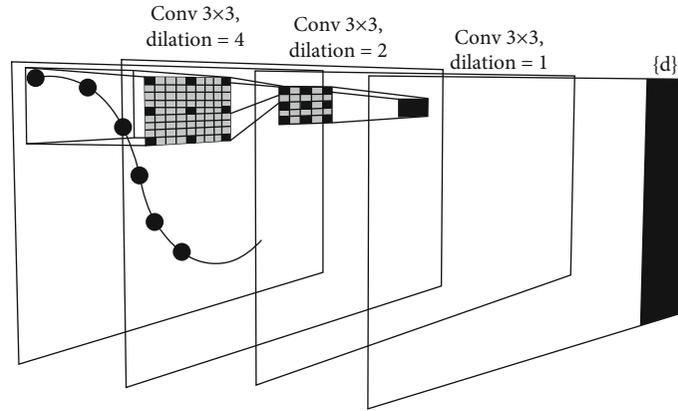


FIGURE 8: Dilation convolution process.

for 2. The temporal prediction sequence $\{h_0, h_1, \dots, h_k\}$ is the output of the TCN.

3.4. Estimation Block. In order to improve the accuracy of blind trajectory prediction, we use a local-global estimation [25] block. The purpose of using the local estimation method is to add the information of the next location point for the blind current location point to form a contextual multiscale calculation process. Therefore, each position point of the blind trajectory will go through the process of local estimation [21]. Compared with the trajectory of a normal person, the position of the blind trajectory is denser per unit time. When multiple local estimations are performed, their cumulative errors have a certain degree of influence on the final prediction results. Therefore, we use global estimation to

reduce the cumulative error of local estimation and improve the global prediction accuracy.

3.5. Local Estimation. After the modeling process of the spatial convolution block and the temporal convolution block, a multidimensional matrix containing temporal and spatial features is formed. In the process of local estimation, the initial 128 dimensions of each h_k are linearly transformed into 1 dimension. An activation function should be performed after each linear transformation. The purpose is to reduce the noise position points in the blind heart trajectory. We take the Leaky ReLU function that has the advantage of a negative saturation region to make the data more inclined to be saturated in the negative region rather than completely return to zero, as shown in Figure 10.

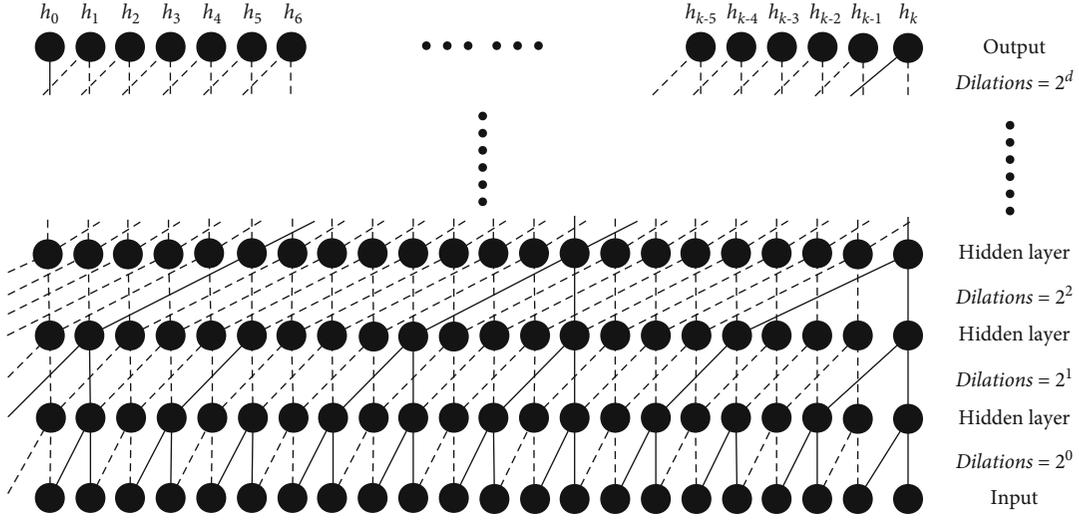


FIGURE 9: Temporal convolution block design.

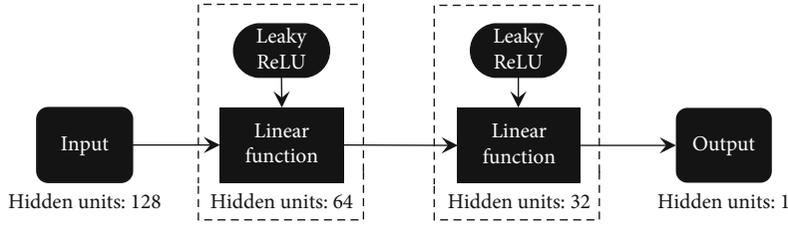


FIGURE 10: Local estimation design.

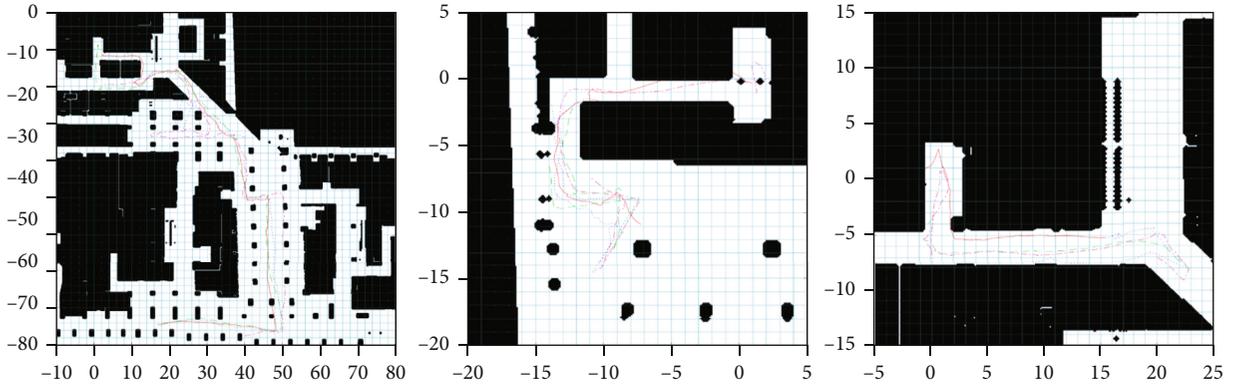


FIGURE 11: The distribution of the trajectory of the blind in the multistorey shopping mall.

3.6. *Global Estimation.* We build a three-layer residual network to reduce the cumulative error of local estimation [21]. We fuse the distance features of the first and the nonfirst nodes of every trajectory with the weights. The feature fusion process of each location point will be realized in a three-layer residual network by layer jump connection. The formula is as

$$h_i = \sigma(h_{i-1} + \text{re}(i)). \quad (10)$$

The $\text{re}(i)$ is the i th residual process, and the h_{i-1} is the result of the previous residual process. The input of the residual unit is directly combined with the output of the residual unit. After that, we use the full connection layer to calculate

the weight of local estimation and residual network features. The global estimation calculates the weight of a specified number of nodes for each blind trajectory that we need to predict.

4. Experiment

4.1. *Data Preparation.* The dataset is collected in a multi-storey shopping mall with three routes in total [26]. It provides time-stamped collections and annotated trajectories, as well as related floor plans. The following are the trajectories of 9 participants in the three routes. From Figure 11, we can know that the route range of the blind does not

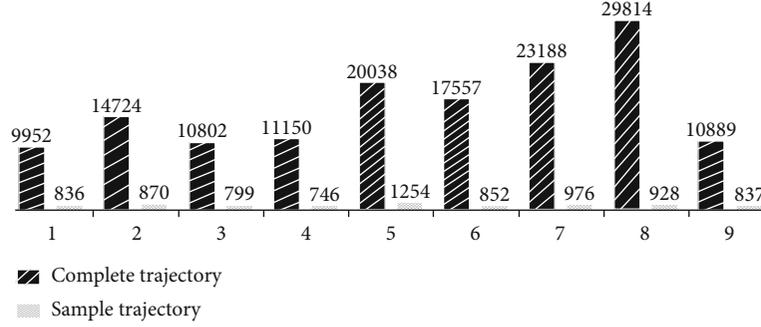


FIGURE 12: The number of position points contained in the original trajectory and sample trajectory of the blind.

include all routes in the three scenes. Therefore, we select part of each scene for experimental modeling. The area we selected contains the entire track range of the 9 participants.

As mentioned earlier, we sample the original data based on the slow trajectory speed of the blind and their insignificant position change per unit time. The sample trajectory represents the long-term timing characteristics of the trajectory. Figure 12 shows the relationship of position node between the complete and sample trajectories of the 9 participants. The time step is 2~3 s in the complete trajectory, but 100~1000 s in the sample trajectory.

4.2. Performance Evaluation

4.2.1. Baselines. In order to verify the performance of the model under the premise of ensuring the scientificity of the results, we use traditional mathematical-statistical models [3–6], traditional convolutional networks, traditional recurrent networks, multilayer dilation convolutional networks, TCN, and complex spatial-temporal models to conduct comparative experiments. The details of the complex spatial-temporal models are shown as follows:

- (i) STF-RNN [27]. It uses a look-up table layer to capture the mixed features of the trajectory in space and time. It inputs this feature into the RNN in an appropriate internal representation method for recursive derivation
- (ii) Social-LSTM [12]. It designs a “social” pooling structure to share the parameter hidden state of the end sequence of the LSTMs. The advantage of this design is that the model can automatically learn the interactions that occur among time-coinciding trajectories
- (iii) Social Attention [13]. It uses a special Structural RNN (S-RNN) to calculate the weight of the spatial-temporal graph data. It takes the problem content as the node and the time series data as the edge value
- (iv) DSCMP [28]. It designs a queue mechanism to explicitly memorize and learn the correlation among long trajectories. It focuses on and uses the consistency characteristics of space and time to capture the contextual parameters of the motion scene

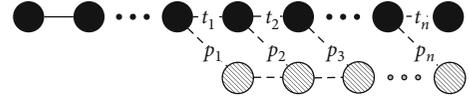


FIGURE 13: Distance error calculation method.

4.2.2. Coordinate. The training in the experiment uses 4 NVIDIA Tesla V100, and the experimental results are the average value after 200 epochs of training under the premise of consistent datasets. The optimization algorithm selects the Adam algorithm, because the learning rate of each iteration has a certain range after bias correction of the Adam algorithm, which makes the parameters relatively stable. We design the corresponding method of error calculation considering the particularity of the dataset, as shown in Figure 13. The $\{d_1, d_2, \dots, d_n\}$ is the sequence of distance which is from real points, and $\{p_1, p_2, \dots, p_n\}$ is the sequence of distance which is between n th predicted points and $(n-1)$ th real points. We calculate the error of distance of these two sequences.

The Root Mean Square Error (RMSE) can measure the deviation between the predicted value and true value. It is usually used as an indicator to measure the prediction accuracy of a deep learning model. The calculation process of RMSE is as shown in Formula (11). We can know that the BlindTPM performs best in the prediction process of medium and long term from Table 1. Although the Markov has the smallest prediction error for the first point, it loses advantage in the prediction process of medium and long term.

$$\text{RMSE} = \sqrt{\frac{1}{m} \sum_{i=1}^m (y_i - \hat{y}_i)^2}. \quad (11)$$

4.2.3. Analysis and Expansion. The mathematical-statistical models are aimed at short-term pedestrian actions and goals. The complexity of Markov is low, and its prediction error for the first point in the future is small. However, its error increases linearly compared with other data-driven models as time goes by. The LSTM is a traditional recurrent network, and it can only capture simple time recursive features of location points without other auxiliary data modeling processes. The D-Conv represents a multilayered dilation convolution model. The translation process of the convolution kernel

TABLE 1: RMSE comparison results.

Position number	Markov	LSTM	D-Conv	STF-RNN	Social-LSTM	Social attention	DSCMP	BlindTPM
1	0.67	3.92	3.32	1.98	1.86	1.42	1.43	1.21
2	2.45	6.32	4.10	2.75	2.76	2.43	2.34	2.01
3	4.87	7.63	5.87	4.86	4.09	3.21	3.23	2.91
4	6.29	8.34	6.55	5.99	5.34	4.79	4.87	4.32
5	8.02	7.91	7.89	6.48	6.02	5.88	5.98	5.23

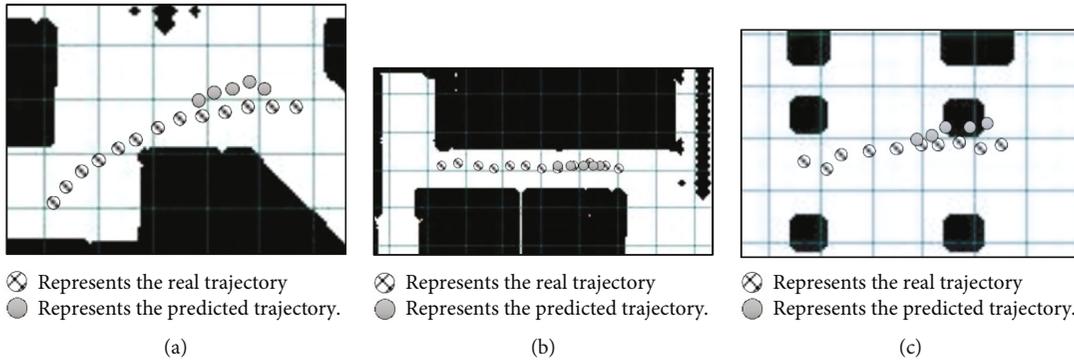


FIGURE 14: The results of BlindTPM prediction results in the coordinate system.

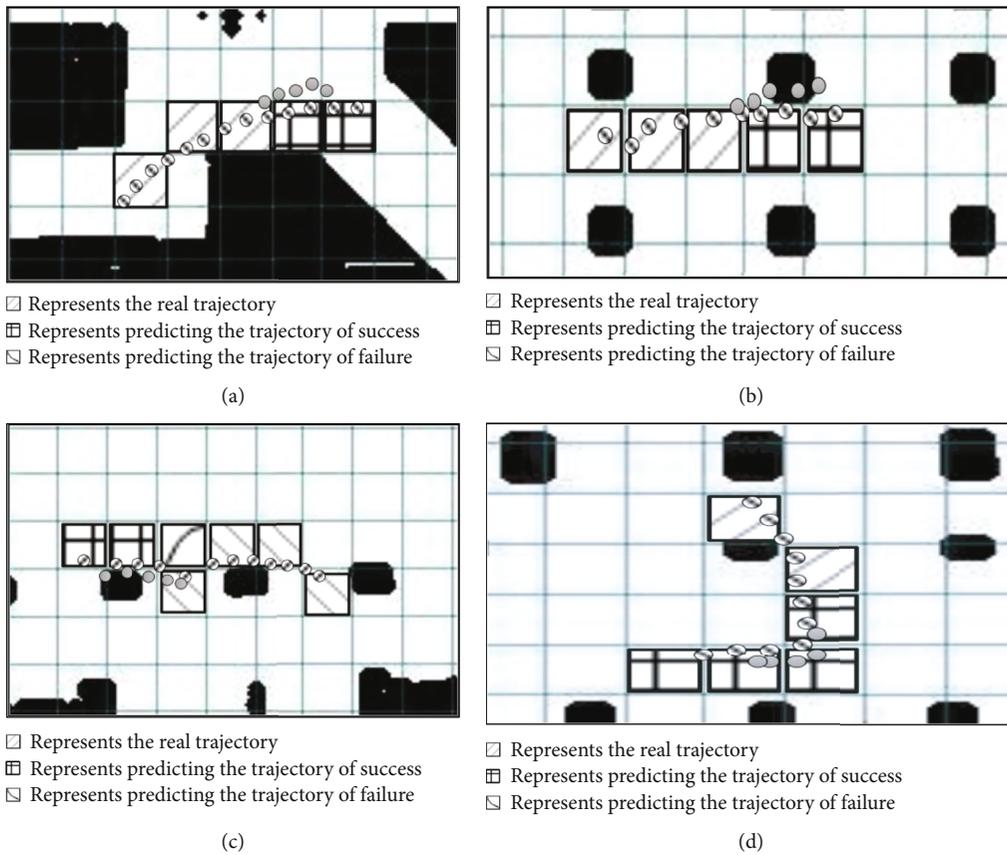


FIGURE 15: The results of BlindTPM prediction results in the obstacle distribution grid.

realizes the feature capture of the trajectory from the bottom to the top. The constantly changing size of the convolution kernel realizes the process of capturing spatial features with

dynamic range changes. However, the nature of D-Conv is a convolution process that makes it cannot perform step-by-step timing-level derivation like recurrent networks. The

TABLE 2: The comparison results of prediction error.

Indicators	Markov	LSTM	D-Conv	STF-RNN	Social-LSTM	Social attention	DSCMP	BlindTPM (no estimation block)	BlindTPM
Accuracy	60%	72%	77%	82%	82%	83%	83%	84%	94%
Precision	59%	73%	80%	83%	79%	81%	84%	83%	93%
Recall	61%	71%	79%	81%	80%	80%	85%	84%	92%

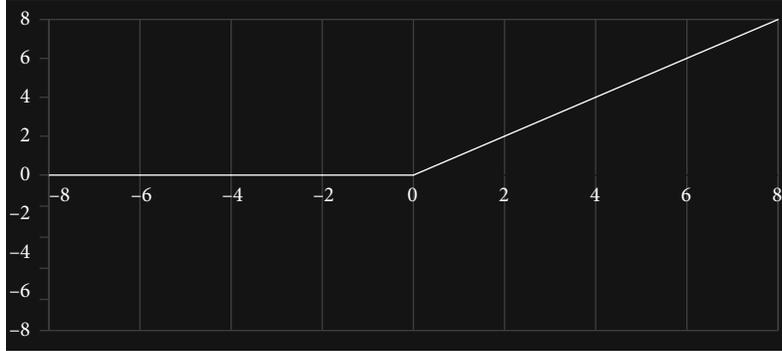


FIGURE 16: Activation function curve. The abscissa is the input value, and the ordinate corresponds to the output value of the input value. The output value of the negative interval is equal to 0, and the output value of the positive interval is the original value.

BlindTPM and other spatial-temporal models consider both the spatial correlation and temporal dependence of the trajectory. Therefore, they have shown a clear advantage in the forecast indicators of the next five points. The BlindTPM uses the same structure as D-Conv in the spatial correlation feature, which can dynamically obtain the spatial distribution weight of the historical trajectory. In the calculation process of temporal dependence, the BlindTPM can use all the historical information of the previous hidden layer to derive the parameter hidden state of the next layer.

Although the above model shows certain data advantages in RMSE, the predicted trajectory has the defects of dense location points and unavoidable obstacles in the actual scene, as shown in Figure 14. Figure 14(a) is an ideal prediction result, and its predicted value has the movement trend of the original trajectory. More importantly, its predicted value is scattered and does not appear in the obstacle area. (b) shows the prediction of a model of a straight trajectory, and it has the defect of too dense predictions. The reason for this phenomenon is that the spatial sample of the linear trajectory is single, which makes the weight of model parameters small. The predicted trajectory in (c) overlaps with the obstacle, because the modeling process does not consider the spatial distribution of the obstacle in the indoor scene. The experimental results of Figure 14 show that there are loopholes in the actual application effect of blind trajectory prediction based on coordinates. We should design additional methods to achieve decentralization of trajectory points to improve the accuracy of prediction. At the same time, we should take the spatial distribution of obstacles in the actual scene as an important reference for the trajectory generation process.

4.2.4. Obstacle Distribution Grid. Figure 15 shows the prediction results of the blind trajectory after the abscissa and ordinate are transformed into grid labels. More importantly, the above results add the spatial distribution of obstacles. (a)

shows that this design mode improves the prediction accuracy and its prediction results are consistent with the real trajectory. (b) and (c) show that the model can avoid obstacles after adding the obstacle distribution. The predicted trajectory points do not penetrate obstacles. (d) shows that the model overcomes the shortcomings of the concentration of prediction results when the data is input into the model in the form of grid labels. The grid at the last point of the blind position is successfully obtained, and the prediction range of the model is broadened. Table 2 is the statistical results of the prediction error of the experimental models. The accuracy indicator represents the percentage of the correct results in the total sample. Although the accuracy indicator can judge the total accuracy rate, it cannot be used as a good indicator to measure the result when the sample is unbalanced. The precision indicator represents the probability of the samples that are actually positive among all the samples that are predicted to be positive. The precision indicator refers to the accuracy of the model's prediction of the results of positive samples, but the accuracy indicator refers to the overall prediction accuracy, which includes both positive samples and negative samples. The recall indicator represents the probability that a positive sample is predicted to be a positive sample. The higher the recall, the higher the probability that the actual negative sample will be predicted. Three indicators are aimed at the average value of the predicted 5 locations. We emphasize that the prediction and evaluation standard of accuracy is that only the grid labels of the abscissa and the ordinate can predict success at the same time. The BlindTPM has an excellent performance in three indicators and has a great improvement in accuracy (11%), precision (9%), and recall (7%) compared to the best performing model (DSCMP).

4.3. Analysis and Expansion. Mapping the coordinate system to grid labels allows us to get rid of the problem of insufficient

predictive ability of the neural network model for floating-point numbers, which also benefits from the invariance of indoor scenes. The grid label makes the trajectory prediction task transformed from a regression problem to a classification problem. The future location is composed of labels with a certain range, and the training process of the model has a clear goal to improve the accuracy of prediction. In the obstacle distribution grid, the area where the obstacle is located is assigned the value 0 and the passable area is assigned the value 1. The GCN weights the obstacle distribution grid and the trajectory distribution grid in the same dimension. After the activation function is calculated, the characteristics of the passable area are enhanced, and the zero-value characteristics of the obstacle area are discarded by the activation function. The activation function (ReLU) makes the feature of the passable area strengthen and the zero-value feature of the obstacle area is discarded, as shown in Figure 16. In order to verify the effect of our proposed estimation block, we set up an ablation experiment, as shown in Table 2. The prediction accuracy of BlindTPM has been significantly improved after adding the estimation block. This is because the local estimation increases the recursive correlation among the trajectory points of the blind. The temporary hidden state is generated by time series data through local estimation. These hidden states that can represent the blind person's short-term future location points are globally estimated to form high-level features that can represent the complete trajectory of the blind. Finally, the estimation block completes the task of improving the prediction accuracy.

5. Conclusions

We propose an obstacle avoidance algorithm and a trajectory prediction spatial-temporal model for the blind aiming at the auxiliary motion tasks of the blind in indoor scenes. Indoor scenes are transformed into grid data in this work. The trajectory of the blind person is accordingly transformed into grid labels. This conversion method limits the range of predicted values and overcomes the difficulty of predicting floating-point trajectory data by neural network models. We convert the obstacles in the scene into the obstacle spatial distribution matrix according to certain rules to reduce the weight of the area where the obstacle is located and increase the weight of the passable area. Experiments verify the difference between coordinate trajectory and grid trajectory, which proves the superiority of the design mode of this work. In future work, we expect that the influence of other populations on the trajectory of blind people will be added to existing research. Its necessity is reflected in the fact that humans are thinking creatures and their actions are always affected by the surrounding environment [29] or other humans to act accordingly.

Data Availability

The dataset of this work comes from the research of Kacorri et al. [26] which is about the movement of blind people indoors. The URL of the original dataset is at <https://envfactors.github.io/>. We perform desensitization processing and obstacle grid processing on the original dataset.

Conflicts of Interest

The authors declare that there is no conflict of interest regarding the publication of this paper.

Acknowledgments

This research was supported in part by the National Key Research and Development Plan Key Special Projects under Grant No. 2018YFB2100303, Shandong Province Colleges and Universities Youth Innovation Technology Plan Innovation Team Project under Grant No. 2020KJN011, Shandong Provincial Natural Science Foundation under Grant No. ZR2020MF060, Program for Innovative Postdoctoral Talents in Shandong Province under Grant No. 40618030001, National Natural Science Foundation of China under Grant No. 61802216, and Postdoctoral Science Foundation of China under Grant No. 2018M642613.

References

- [1] M. Xiao, J. Wu, L. Huang, R. Cheng, and Y. Wang, "Online task assignment for crowdsensing in predictable mobile social networks," *IEEE Transactions on Mobile Computing*, vol. 16, no. 8, pp. 2306–2320, 2017.
- [2] L. Li, B. Zhou, J. Lian, and Y. Zhou, "Research on pedestrian trajectory prediction method based on social attention mechanism," *Journal on Communications*, vol. 41, no. 6, pp. 175–183, 2020.
- [3] D. Helbing and P. Molnar, "Social force model for pedestrian dynamics," *Physical Review E*, vol. 51, no. 5, pp. 4282–4286, 1995.
- [4] P. Trautman and A. Krause, "Unfreezing the robot: navigation in dense, interacting crowds," in *2010 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 797–803, Taipei, Taiwan, October 2010.
- [5] B. T. Morris and M. M. Trivedi, "Trajectory learning for activity understanding: unsupervised, multilevel, and long-term adaptive approach," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 33, no. 11, pp. 2287–2301, 2011.
- [6] K. M. Kitani, B. D. Ziebart, J. A. Bagnell, and M. Hebert, "Activity forecasting," in *Computer Vision – ECCV 2012. ECCV 2012. Lecture Notes in Computer Science, vol 7575*, A. Fitzgibbon, S. Lazebnik, P. Perona, Y. Sato, and C. Schmid, Eds., pp. 201–214, Springer, Berlin, Heidelberg, 2012.
- [7] P. Dayan and B. W. Balleine, "Reward, motivation, and reinforcement learning," *Neuron*, vol. 36, no. 2, pp. 285–298, 2002.
- [8] Z. Y. Zhang and Y. H. Diao, "Pedestrian trajectory prediction model with social features and attention," *Journal of Xidian University (Natural Science)*, vol. 47, no. 1, pp. 10–17, 2020.
- [9] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Computation*, vol. 9, no. 8, pp. 1735–1780, 1997.
- [10] K. Xu, Z. Qin, G. Wang, K. Huang, S. Ye, and H. Zhang, "Collision-free lstm for human trajectory prediction," in *MultiMedia Modeling. MMM 2018. Lecture Notes in Computer Science, vol 10704*, pp. 106–116, Springer, Cham, 2018.
- [11] J. Liu, A. Shahroudy, D. Xu, and G. Wang, "Spatio-temporal LSTM with trust gates for 3D human action recognition," in *Computer Vision – ECCV 2016. ECCV 2016. Lecture Notes in Computer Science, vol 9907*, B. Leibe, J. Matas, N. Sebe, and M. Welling, Eds., pp. 816–833, Springer, Cham, 2016.

- [12] A. Alahi, K. Goel, V. Ramanathan, A. Robicquet, L. Fei-Fei, and S. Savarese, "Social LSTM: human trajectory prediction in crowded spaces," in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 961–971, Las Vegas, NV, USA, June 2016.
- [13] A. Vemula, K. Mueller, and J. Oh, "Social attention: modeling attention in human crowds," in *2018 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 4601–4607, Brisbane, QLD, Australia, May 2018.
- [14] V. Todi, G. Sengupta, and S. Bhattacharya, "Probabilistic path planning using obstacle trajectory prediction," in *Proceedings of the ACM India Joint International Conference on Data Science and Management of Data.*, pp. 36–43, Kolkata India, 2019.
- [15] Z. Wang, M. L. Fang, and X. N. Lin, "Route planning in handheld blind navigation system based on RFID," *Computer Engineering and Design*, vol. 33, no. 5, pp. 2063–2067, 2012.
- [16] M. Luo, X. Hou, and J. Yang, "Surface optimal path planning using an extended Dijkstra algorithm," *IEEE Access*, vol. 8, no. 1, pp. 147827–147838, 2020.
- [17] F. Wu, A. Souza, T. Zhang, C. Fifty, T. Yu, and K. Weinberger, "Simplifying graph convolutional networks," in *Proceedings of the 36th International Conference on Machine Learning*, pp. 6861–6871, 2019.
- [18] K. O'Shea and R. Nash, "An introduction to convolutional neural networks," 2015, <http://arxiv.org/abs/1511.08458>.
- [19] L. Zhao, Y. Song, C. Zhang et al., "T-gcn: a temporal graph convolutional network for traffic prediction," *IEEE Transactions on Intelligent Transportation Systems*, vol. 21, no. 9, pp. 3848–3858, 2020.
- [20] W. Sun and X. Zhou, "Characterization of local sampling sequences for spline subspaces," *Advances in Computational Mathematics*, vol. 30, no. 2, pp. 153–175, 2009.
- [21] Z. Lv, J. Li, C. Dong, and W. Zhao, "A deep spatial-temporal network for vehicle trajectory prediction," in *Wireless Algorithms, Systems, and Applications. WASA 2020. Lecture Notes in Computer Science, vol 12384*, D. Yu, F. Dressler, and J. Yu, Eds., pp. 359–369, Springer, Cham, 2020.
- [22] D. K. Hammond, P. Vandergheynst, and R. Gribonval, "Wavelets on graphs via spectral graph theory," *Applied and Computational Harmonic Analysis*, vol. 30, no. 2, pp. 129–150, 2011.
- [23] M. Defferrard, X. Bresson, and P. Vandergheynst, "Convolutional neural networks on graphs with fast localized spectral filtering," 2016, <http://arxiv.org/abs/1606.09375>.
- [24] S. Bai, J. Z. Kolter, and V. Koltun, "An empirical evaluation of generic convolutional and recurrent networks for sequence modeling," 2018, <http://arxiv.org/abs/1803.01271>.
- [25] D. Wang, J. Zhang, W. Cao, J. Li, and Y. Zheng, "When will you arrive? Estimating travel time based on deep neural networks," *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 32, 2018.
- [26] H. Kacorri, E. Ohn-Bar, K. M. Kitani, and C. Asakawa, "Environmental factors in indoor navigation based on real-world trajectories of blind users," in *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems*, pp. 1–12, Montreal QC Canada, 2018.
- [27] A. Al-Molegi, M. Jabreel, and B. Ghaleb, "STF-RNN: space time features-based recurrent neural network for predicting people next location," in *2016 IEEE Symposium Series on Computational Intelligence (SSCI)*, pp. 1–7, Athens, Greece, December 2016.
- [28] C. Tao, Q. Jiang, L. Duan, and P. Luo, "Dynamic and static context-aware LSTM for multi-agent motion prediction," pp. 547–563, 2020, <http://arxiv.org/abs/2008.00777>.
- [29] K. Mangalam, H. Girase, S. Agarwal et al., "It is not the journey but the destination: endpoint conditioned trajectory prediction," in *Proceedings of the 16th European Conference on Computer Vision*, pp. 759–776, Glasgow, UK, 2020.