IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received February 12, 2020, accepted February 26, 2020, date of publication March 9, 2020, date of current version March 19, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.2979562

Panoramic Camera-Based Human Localization Using Automatically Generated Training Data

YONGLIANG SUN[1], (Member, IEEE), WEIXIAO MENG[2], (Senior Member, IEEE), CHENG LI[3], (Senior Member, IEEE), AND XUZI WU[1]

[1]School of Computer Science and Technology, Nanjing Tech University, Nanjing 211816, China
[2]School of Electronics and Information Engineering, Harbin Institute of Technology, Harbin 150001, China
[3]Department of Electrical and Computer Engineering, Faculty of Engineering and Applied Science, Memorial University, St. John's, NL A1B 3X5, Canada

Corresponding author: Yongliang Sun (syl_peter@163.com)

This work was supported in part by the National Natural Science Foundation of China under Grant 61701223 and in part by the Natural Science Foundation of Jiangsu Province under Grant BK20171023.

ABSTRACT In this paper, a panoramic camera-based human localization method using automatically generated training data is proposed to locate a human target accurately in a room scenario. The method recognizes a feature object and detects the edge pixel locations of the object in the observed image and room layout map. Then it partitions the target area into four subareas and matches the edge pixel locations of each subarea in the image with the ones in the layout map to generate the training data. A training data augmentation method is also proposed to automatically generate quadruple training data for localization performance improvement. With the generated training data, general regression neural network (GRNN) is used to construct one regression model for each subarea to calculate the human target's location. When the human target is observed and detected as a foreground target in the image, the foreground pixel location that can represent the human target's location most accurately is searched and used to calculate the human target's location coordinates with one of the four constructed GRNN models. Experimental results demonstrate that our panoramic camera-based human localization method is able to achieve a mean error of 0.77m, which outperforms fingerprinting and propagation model localization methods.

INDEX TERMS Human localization, panoramic camera, general regression neural network, training data generation.

I. INTRODUCTION

With the development of mobile communications and internet of things (IoT), the demands of location-based service (LBS) increase rapidly because various applications require LBS to work effectively and efficiently, such as electronic commerce, health care, social network, and emergency response [1]–[3]. Because the performance of global navigation satellite system (GNSS) and cellular network localization could be limited in indoor environments [4]–[6], many indoor localization methods have been proposed and developed using radio frequency identification (RFID), ultra-wideband (UWB), ZigBee, Wi-Fi, inertial sensors and so on [7]–[12]. Most of these localization methods require people to take terminal devices, which may be not applicable to some application scenarios like the localization and tracking for patient rehabilitation supervision [13]. As the rapid proliferation of cameras in people's daily life for security and surveillance applications, cameras have been an essential part of IoT [14]. Moreover, sometimes people might be interested not only in the existence of some observed targets but also in the location information of these targets [15].

Therefore, camera-based human localization and tracking have attracted extensive research interests and played an important role in IoT. Most of the exiting camera-based localization methods utilize multiple cameras to monitor a wide area due to the limited field of views, which also involves some problems like the deployment and cooperation of these cameras. As we mentioned in our previous work [5], because a panoramic camera is able to cover a wide angle, one panoramic camera can be used to monitor an open area like an office room. Although multiple cameras might be still needed in the scenarios that are not completely open, the use of panoramic camera is able to reduce the number of cameras to some degree. Therefore, we propose a panoramic camera-based human localization method using automatically generated training data in this paper. In [5], we constructed a regression model for human

The associate editor coordinating the review of this manuscript and approving it for publication was Li He.

localization using a basic multi-layer perceptron (MLP) trained by back-propagation (BP) algorithm. We derived the training data for the MLP through taking videos of a person with known locations, which was a time-consuming and laborious process. By comparison, our proposed method is able to generate the training data automatically and augment the training data for performance improvement. Moreover, general regression neural network (GRNN) that has a superior performance compared with the MLP is used as the regression model for human localization using the generated training data. The main contributions of this paper are summarized as follows:

1) We propose a panoramic camera-based human localization method using automatically generated training data, which has a superior localization performance. The method is able to not only locate a human target without any terminal device, but also generate training data automatically.

2) We propose a training data generation method. The method first recognizes a feature object that is the maximum object in the image plan and layout map and detects the edge pixel locations of the feature object. Then it partitions the target area into four subareas and matches the edge pixel locations of each subarea in the image with the ones in the layout map to generate the training data.

3) We propose a training data augmentation method to improve the localization performance. The method transforms the edge pixel locations to one subarea from the ones of the other three subareas in turn. More training data can be obtained and used to construct one regression model for each subarea. Also, the target area of each GRNN model is reduced to one fourth of the whole area, which is beneficial to performance improvement.

4) We verify the proposed panoramic camera-based human localization method in a real room scenario and compare the proposed localization method with fingerprinting and propagation model (PM) localization methods. We also compare the GRNN model with other popular regression models for calculating the human target's location. The experimental results demonstrate that our proposed localization method has a superior performance.

The remainder of this paper is organized as follows. Section II reviews the related works of the proposed panoramic camera-based localization method. In Section III, the overview of the proposed localization method is described and the details of each component of the method are given. The experimental setup, results, and analyses are presented in Section IV. Finally, Section V concludes the whole paper.

II. RELATED WORKS

So far, many camera-based localization methods have been proposed. Liu *et al.* [15] first defined cost and utility functions to balance the tradeoff between the localization accuracy and energy cost. They also optimized the selected subset of cameras for calculating the target's location. After that, they focused on the camera coverage problem in [16]. They proposed a localization-oriented sensing model and analyzed the relationship between the camera density and localization coverage probability. Lobaton *et al.* [17] presented a camera network representation called camera network complex for tracking applications, which accurately captured topological information about network coverage. This representation was effective in tracking targets. Reference [18] introduced a Bayesian approach on people localization using multiple cameras. Features from these cameras were fused to create evidence for the location and height of a person. With this information, a cylinder object was used to approximate the person's location. Liu *et al.* [19] presented a location-constrained maximum *a posteriori* algorithm for camera-based localization using camera parameters and known location information. A task-oriented evaluation metric was also presented by them to evaluate the localization results. Lin *et al.* [20] proposed a series of image transforms using the vanishing point of vertical lines for the enhancement of probabilistic occupancy map-based people localization.

Meanwhile, camera-based localization and tracking could also be assisted with laser projectors, vehicles and so on. Lu *et al.* [21] presented an image-based framework for measuring targets on an oblique plane using a camera and two laser projectors. They first measured the photographic distance and then used their framework to locate objects on a ground surface. In [22], Miseikis *et al.* proposed an approach that integrated information from static cameras and a mobile camera mounted on a vehicle. They applied background subtraction and histograms of oriented gradients to detect people in the static and mobile camera images, respectively. They tracked people through combining the outputs of the static and mobile cameras. Minaeian *et al.* [23] presented a vision-based target detection and localization system using an unmanned aerial vehicle (UAV) and multiple unmanned ground vehicles (UGVs). A motion detection algorithm was applied to follow people using the camera mounted on the UAV. The UGVs were used as human detectors and moving landmarks for human localization. Also, with cameras, Raspberry Pi boards, and Kinect devices, a wireless camera sensor network platform consisted of three camera sensors was established in [13]. The platform could meet the localization and tracking requirements for patient rehabilitation supervision.

Compared with the above-mentioned camera-based localization methods using multiple cameras or a single camera assisted with extra equipment, our proposed method utilizes one panoramic camera to monitor a target area like an office room and is able to generate training data automatically to construct the regression models for human localization.

III. PROPOSED PANORAMIC CAMERA-BASED HUMAN LOCALIZATION METHOD

A. METHOD OVERVIEW

The proposed panoramic camera-based human localization method consists of three main components: target pixel location search, training data generation, and regression model construction. Among them, the training data generation

component includes feature object recognition, edge detection, area partition, pixel location matching, and training data augmentation.

In target pixel location search, the images recorded by the panoramic camera are processed and a human target can be detected as a foreground target with background subtraction method [24]. The foreground pixel location that is able to represent the human target's pixel location most accurately is searched and considered as the human target's pixel location in the observed image.

In training data generation, we detect all the connected components in the binary image derived from the observed image and label them with different digits to recognize the maximum object in the image plan. Then the edge pixels of the maximum object in the image can be detected with Canny edge detection algorithm [25]. Meanwhile, the edge pixels of the same maximum object in the layout map can be detected in the same way only using the location and outline information of the maximum object in the layout map.

According to the pixel location of the panoramic camera, we partition the target area into four subareas. We match the edge pixel locations of each subarea in the image with the ones in the layout map. So the pixel location data in the image can be used as the inputs of one regression model and the corresponding pixel location data in the layout map can be the outputs of the model. We also transform the edge pixel locations to one subarea from the ones of the other three subareas in turn to obtain quadruple edge pixel location data for localization performance improvement.

With the edge pixel location data, we construct one regression model for each subarea to compute the human target's pixel location in the layout map. When a human target is observed by the camera, the human target's pixel location data in the image can be computed and input into one regression model. Then the human target's location coordinates in the layout map are computed by the regression model and transformed to be the localization coordinates in our constructed coordinate system with two linear functions.

B. TARGET DETECTION AND PIXEL LOCATION SEARCH

Human target detection is the prerequisite of the proposed panoramic camera-based human localization method. In this paper, we detect a human target through separating the human target, namely foreground target, from the modeled background image using the basic background subtraction method, which is a widely used target detection method. Firstly, let $\mathbf{I}=\{\mathbf{i}_1, \mathbf{i}_2, \cdots, \mathbf{i}_t\}$ denote an RBG image set obtained from the recorded videos. After image processing operations including resizing, rotation, gray scale processing, and reverse color processing, we can get a gray scale image set $\mathbf{F}=\{\mathbf{f}_1, \mathbf{f}_2, \cdots, \mathbf{f}_t\}$ and any pixel value of these gray scale images ranges from 0 to 255. Therefore, the background image can be modeled through:

$$b_t(x_i, y_i) = \frac{1}{L}\sum_{k=0}^{L-1} f_{t-k}(x_i, y_i), \tag{1}$$

where $b_t(x_i, y_i)$ is the pixel value at location (x_i, y_i) in the background image $\mathbf{b}_t$, $f_{t-k}(x_i, y_i)$ is the pixel value at location (x_i, y_i) in the image $\mathbf{f}_{t-k}$, L is the number of images used for computing $b_t(x_i, y_i)$. After all the background pixel values are computed, we can get the background image $\mathbf{b}_t$. Let $f_t(x_i, y_i)$ be the pixel value at location (x_i, y_i) in the current gray scale image $\mathbf{f}_t$, then we compute the difference value between $b_t(x_i, y_i)$ and $f_t(x_i, y_i)$ for detecting a foreground pixel with:

$$|b_t(x_i, y_i) - f_t(x_i, y_i)| > T, \tag{2}$$

where T is a threshold that is utilized to determine whether the pixel at location (x_i, y_i) belongs to the foreground target or not. If (2) is fulfilled, then the pixel is defined as a foreground pixel, or it belongs to the background.

After the target detection, let $\mathbf{Q}=\{\boldsymbol{q}_1, \boldsymbol{q}_2, \cdots, \boldsymbol{q}_l\}$ be the set of the detected foreground pixel locations, where $\boldsymbol{q}_i = (x_i^q, y_i^q)$, $i \in \{1, 2, \cdots, l\}$. Then we search the pixel location that can represent the human target's location most accurately. After intensive research, we study that the foreground pixel location in set $\mathbf{Q}$ that is the nearest to the pixel location of the panoramic camera $\boldsymbol{r}_{\mathrm{P}} = (x_{\mathrm{P}}, y_{\mathrm{P}})$ can represent the human target's location most accurately and therefore can be considered as the target's location. To search this foreground pixel location, we calculate the Euclidean distance d_i, $i \in \{1, 2, \cdots, l\}$ between the foreground pixel location $\boldsymbol{q}_i$, $i \in \{1, 2, \cdots, l\}$ and the pixel location of the panoramic camera $\boldsymbol{r}_{\mathrm{P}}$ and then select the foreground pixel location $\boldsymbol{q}_n$ with the minimum distance d_n as the human target's location $\boldsymbol{q}_{\mathrm{Im}}$ in the image, which can be denoted by:

$$\begin{cases} d_n = \min\left\{\left\|\boldsymbol{r}_{\mathrm{P}} - \boldsymbol{q}_i\right\|_2\right\}, \quad i \in (1, 2, \cdots, l) \\ \boldsymbol{q}_{\mathrm{Im}} = \boldsymbol{q}_n, \end{cases} \tag{3}$$

where $\|\cdot\|_2$ is the l_2-norm.

C. EDGE DETECTION AND AREA PARTITION

1) EDGE DETECTION

We first binarize the background image and then perform morphological operations in order to recognize independent objects accurately in the binary image. Using connected component labeling [26], proximity relationship among the pixels can be evaluated to determine whether a pixel belongs to an already recognized object or not. So the connected components can be labeled with different digits. Then the maximum connected component, namely the maximum object, can be recognized in the observed image plan. We detect the edge of the maximum object with Canny edge detection algorithm. The main steps of the algorithm are as follows:

(1) Image Smoothing

In order to restrain noise, the input image is smoothed by a Gaussian filter and the Gaussian filter outputs $G(x, y)$.

(2) Gradient Magnitude and Direction Calculation

The algorithm calculates the horizontal derivative G_x and vertical derivative G_y. Then the gradient magnitude M and direction θ are calculated with G_x and G_y.

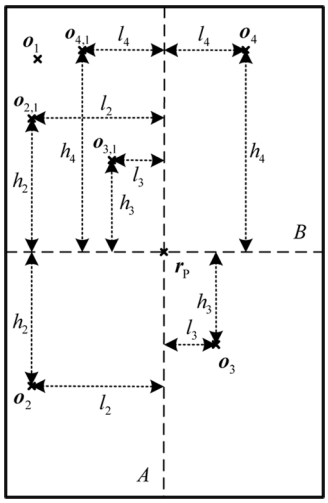


FIGURE 1. Training data augmentation through transforming pixel locations to one subarea from the ones in the other three subareas.

(3) Non-Maximum Suppression

In order to thin the edge, the non-maximum suppression of Canny edge detection algorithm is able to suppress all the non-maximum gradient values except the local maximum one. The local maximum gradient value denotes a pixel location with the sharpest change of intensity value.

(4) Edge Extraction

Through empirically selecting the high and low gradient thresholds, Canny edge detection algorithm filters out the edge pixels with a low gradient value and preserves the edge pixels with a high gradient value to determine the final edge of the object in the image.

Meanwhile, we also measure the location and outline of the maximum object in the target area and draw an accurate layout map. The edge of the maximum object in the layout map can also be detected using the same algorithm.

2) AREA PARTITION

After we get the edge pixels of the maximum object in the observed image, we assume that the set of these edge pixel locations is denoted by $\mathbf{O}_{\mathrm{Im}}$. Then we partition the target area in the image into four subareas with straight lines A and B that go through the pixel location of the panoramic camera $\boldsymbol{r}_{\mathrm{P}} = (x_{\mathrm{P}}, y_{\mathrm{P}})$ and are parallel to the height and width of the image, respectively, as shown in Fig. 1. Accordingly, the set of the edge pixel locations $\mathbf{O}_{\mathrm{Im}}$ is divided into four sets $\mathbf{O}_i$, $i \in \{1, 2, 3, 4\}$, which belong to the four subareas from the up-left subarea to the up-right subarea in a counter-clockwise order, respectively. We can also assume that the set of the edge pixel locations in the map is denoted by $\mathbf{L}_{\mathrm{Ma}}$ and is divided into four sets $\mathbf{L}_i$, $i \in \{1, 2, 3, 4\}$ using the same area partition method.

D. PIXEL LOCATION MATCHING AND TRAINING DATA AUGMENTATION

1) PIXEL LOCATION MATCHING

According to the result of area partition, we match the edge pixel locations of each subarea in the image with the ones in the layout map to generate training data automatically. Let $\mathbf{O}_i = \{\boldsymbol{o}_{i(1)}, \boldsymbol{o}_{i(2)}, \cdots, \boldsymbol{o}_{i(m)}\}$ and $\mathbf{L}_i = \{\boldsymbol{l}_{i(1)}, \boldsymbol{l}_{i(2)}, \cdots, \boldsymbol{l}_{i(n)}\}$ be the sets of edge pixel locations of the ith subareas in the image and layout map, respectively, where m and n are the numbers of the edge pixel locations in the two sets. For the ith subarea, if m is not equal to n, then we delete the redundant data in the image to obtain a new set denoted by $\mathbf{O}'_i$ or delete the redundant data in the layout map to obtain a new set denoted by $\mathbf{L}'_i$, so that the numbers of the edge pixel locations in the two sets can be equal for matching. If m is equal to n, then we directly match the pixel locations in the two sets in turn. The matched pixel location pairs can be the generated training data. The proposed pixel location matching algorithm is described in Algorithm 1. After matching the edge pixel locations in each subarea, the two edge pixel location sets in the image and layout map can be denoted by $\mathbf{O}'_{\mathrm{Im}}$ and $\mathbf{L}'_{\mathrm{Ma}}$, respectively, and they have the same number of the edge pixel locations.

2) TRAINING DATA AUGMENTATION

As mentioned above, we transform the edge pixel locations to one subarea from the ones in the other three subareas in turn. In Fig. 1, we take four pixel locations $\boldsymbol{o}_i = (x_i, y_i)$, $i \in \{1, 2, 3, 4\}$ in the image as an example, then the horizontal and vertical distances l_i and h_i between $\boldsymbol{r}_{\mathrm{P}}$ and $\boldsymbol{o}_i$ can be calculated by:

$$\begin{cases} l_i = |x_i - x_{\mathrm{P}}| \\ h_i = |y_i - y_{\mathrm{P}}| \,. \end{cases} \tag{4}$$

According to the symmetry, we can obtain the three transformed edge pixel locations $\boldsymbol{o}_{j,1}, j \in \{2, 3, 4\}$ from the down-left, down-right, and up-right subareas to the up-left subarea, respectively. So we can have four edge pixel locations in the up-left subarea shown in Fig. 1. Similarly, after we obtain the sets of edge pixel locations $\mathbf{O}'_i$, $i \in \{1, 2, 3, 4\}$ in the image through the pixel location matching, the sets of the transformed pixel locations from the down-left, down-right, and up-right subareas to the up-left subarea can be calculated and denoted by $\mathbf{O}'_{j,1}, j \in \{2, 3, 4\}$, respectively. In the same way, we can also obtain the sets of edge pixel locations $\mathbf{L}'_i$, $i \in \{1, 2, 3, 4\}$ in the layout map and then the transformed sets from the down-left, down-right, and up-right subareas to the up-left subarea can be calculated and denoted by $\mathbf{L}'_{j,1}, j \in \{2, 3, 4\}$, respectively. So we can have four sets of the edge pixel locations in the up-left area of the image or layout map. Then we transform the corresponding sets of the edge pixel locations to the down-left, down-right, and up-right subareas from the other three subareas, respectively. After the training data augmentation, we can have two new sets of edge pixel locations of the whole area denoted by $\mathbf{O}'_{\mathrm{Im\_New}} = \left\{\mathbf{O}'_i, \mathbf{O}'_{j,i}, i, j \in \{1, 2, 3, 4\}, i \neq j\right\}$ and $\mathbf{L}'_{\mathrm{Ma\_New}} = \left\{\mathbf{L}'_i, \mathbf{L}'_{j,i}, i, j \in \{1, 2, 3, 4\}, i \neq j\right\}$, which are used for constructing the regression models. The number of the edge pixel locations in the new set $\mathbf{O}'_{\mathrm{Im\_New}}$ or $\mathbf{L}'_{\mathrm{Ma\_New}}$ is

Algorithm 1 Algorithm of Matching Edge Pixel Locations of Subareas in the Image and Layout Map

Input:
The edge pixel location sets $\mathbf{O}_i$ and $\mathbf{L}_i$ of the ith subareas in the image and layout map;
Output:
The matched result of the edge pixel location sets $\mathbf{O}'_i$ and $\mathbf{L}'_i$ of the ith subareas in the image and layout map;
Calculating the numbers of edge pixel locations in the two sets m and n;
if $m > n$ then
  calculating $s = [m/(m-n)]$;
  for $k = 0 : (m-n) - 1$ do
    $\boldsymbol{o}_{i,k\cdot s+s} = 0$;
  end for
  Deleting 0 in $\mathbf{O}_i$ and obtaining a new set $\mathbf{O}'_i$;
  $\mathbf{L}'_i = \mathbf{L}_i$;
  Matching pixel locations in $\mathbf{O}'_i$ and $\mathbf{L}'_i$;
else if $m < n$ then
  calculating $s = [n/(n-m)]$;
  for $k = 0 : (n-m) - 1$ do
    $\boldsymbol{l}_{i,k\cdot s+s} = 0$;
  end for
  Deleting 0 in $\mathbf{L}_i$ and obtaining a new set $\mathbf{L}'_i$;
  $\mathbf{O}'_i = \mathbf{O}_i$;
  Matching pixel locations in $\mathbf{O}'_i$ and $\mathbf{L}'_i$;
else
  $\mathbf{O}'_i = \mathbf{O}_i$;
  $\mathbf{L}'_i = \mathbf{L}_i$;
  Matching pixel locations in $\mathbf{O}'_i$ and $\mathbf{L}'_i$;
end if

three times more than the number of the edge pixel locations in $\mathbf{O}'_{\mathrm{Im}}$ or $\mathbf{L}'_{\mathrm{Ma}}$.

E. GRNN MODEL CONSTRUCTION

Because we partition the target area into four subareas, we can construct one GRNN model for each subarea and select one of the four constructed GRNN models for human localization according to the target's pixel location in the image, which also means the coverage area of each GRNN model is reduced to one fourth of the whole area. In order to improve the localization performance still, we calculate the radius $\rho_{i(k)}$ and angle $\theta_{i(k)}$ of the kth pixel location $\boldsymbol{o}_{i(k)} = \left(x_{i(k)}, y_{i(k)}\right)$ in the ith subarea of the image and also used them as the inputs of the GRNN model. The radius $\rho_{i(k)}$ and angle $\theta_{i(k)}$ can be computed by:

$$\begin{cases} \rho_{i(k)} = \sqrt{\left(x_{i(k)} - x_{\mathrm{P}}\right)^2 + \left(y_{i(k)} - y_{\mathrm{P}}\right)^2} \\ \theta_{i(k)} = \arctan \dfrac{\left|y_{i(k)} - y_{\mathrm{P}}\right|}{\left|x_{i(k)} - x_{\mathrm{P}}\right|} \end{cases} \tag{5}$$

We can have an input vector of the GRNN $\boldsymbol{s}_{i(k)}$ that contains the pixel location coordinates $\left(x_{i(k)}, y_{i(k)}\right)$, radius $\rho_{i(k)}$, and angle $\theta_{i(k)}$. We model the relationship between the vector $\boldsymbol{s}_{i(k)}$ in the image and the pixel location vector $\boldsymbol{l}_{i(k)}$ in the layout map using the GRNN denoted by:

$$\boldsymbol{l}_{i(k)} = F_i\left(\boldsymbol{s}_{i(k)}\right). \tag{6}$$

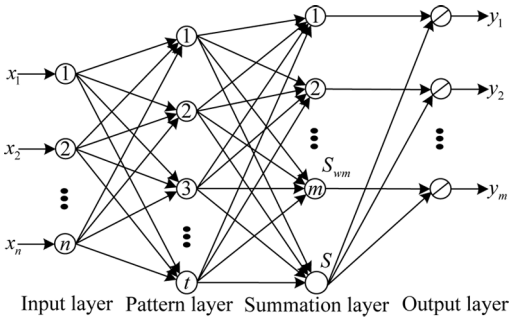


FIGURE 2. The structure of the GRNN model.

Regarding the GRNN, it is a variation of the radial basis function (RBF) neural network and has been widely used for function approximation [27]. A basic GRNN has four layers that are the input layer, the pattern layer, the summation layer, and the output layer shown in Fig. 2. A GRNN does not need an iterative training process as an MLP trained by BP algorithm and also does not require a tremendous number of calculations as deep learning algorithms. Meanwhile, GRNN has a better performance than the MLP or RBF [28] and has only one key parameter to be determined called the spread parameter, which can be calculated through cross validation. Thus, we exploit GRNN for human localization in this paper.

We assume that the input and output vectors of a GRNN are denoted by $\boldsymbol{x}=\{x_1, x_2, \cdots, x_n\}$ and $\hat{\boldsymbol{y}}=\left\{\hat{y}_1, \hat{y}_2, \cdots, \hat{y}_m\right\}$, respectively, and the set of training samples is denoted by $\mathbf{T} = \left\{\boldsymbol{x}_i, \boldsymbol{y}_i, i \in \{1, 2, \cdots, t\}\right\}$. The goal of the GRNN is to compute the output vector $\hat{\boldsymbol{y}}$. The number of neurons in the input layer is the same as the number of inputs and each neuron in the input layer is connected to the neurons in the pattern layer. Each neuron in the pattern layer calculates a Gaussian function denoted by:

$$\begin{cases} g_i = \exp\left(-\dfrac{D_i^2}{2\sigma^2}\right) \\ D_i^2 = (\boldsymbol{x} - \boldsymbol{x}_i)^{\mathrm{T}} (\boldsymbol{x} - \boldsymbol{x}_i), \end{cases} \tag{7}$$

where σ is the spread parameter and D_i^2 is the squared distance between $\boldsymbol{x}$ and $\boldsymbol{x}_i$.

As shown in Fig. 2, the summation layer has two kinds of neurons. The jth neuron in the first kind calculates the weighted sum of the pattern layer outputs S_{wj} and the single one neuron in the second kind calculates the sum of the pattern layer outputs S, then the jth component of the output

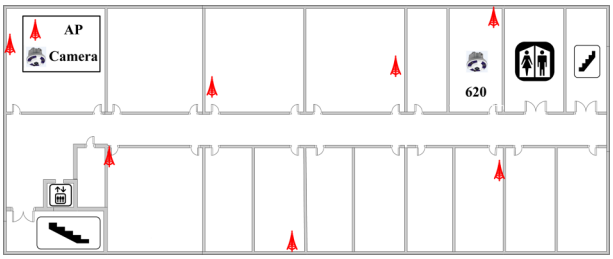
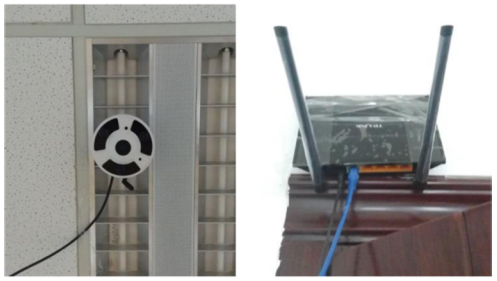


FIGURE 3. The experimental floor plan.



(a) (b)

FIGURE 4. (a) The mounted 28mm panoramic camera, (b) TP-LINK TL-WR845N AP.

vector can be calculated by:

$$\begin{cases} \hat{y}_j = \dfrac{S_{wj}}{S_t} \\ S_{wj} = \displaystyle\sum_{i=1}^{t} y_{ij} g_i \\ S = \displaystyle\sum_{i=1}^{t} g_i, \end{cases} \tag{8}$$

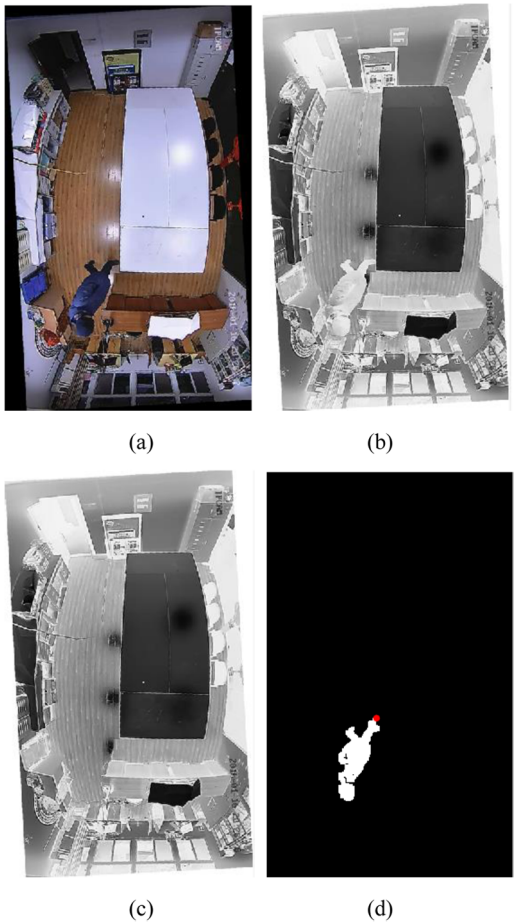
where $\hat{y}_j$ and y_{ij} are jth components of the vectors $\hat{\boldsymbol{y}}$ and $\boldsymbol{y}_i$, respectively. With the constructed GRNN, the human target's location coordinates in the layout map can be calculated.

IV. EXPERIMENTAL SETUP, RESULTS, AND ANALYSES

A. EXPERIMENTAL SETUP

We collected all the experimental data on a typical office floor with dimensions of 51.6m×20.4m×2.7m shown in Fig. 3. As shown in Fig. 4(a), we mounted a 28mm CMOS panoramic camera at the central area of Room 620, which is a rectangle office room with dimensions of 5.1m×8.5m×2.7m. In order to obtain the outputs of a regression model for its construction, we measured the location and outline information of the maximum object in Room 620 for the edge detection. We used 1200 images as the testing data derived from the videos of the human target in Room 620 recorded by the mounted panoramic camera.

For localization performance comparison, we also performed fingerprinting and PM localization methods using Wi-Fi in the same experimental scenario. We deployed 7 TP-LINK TL-WR845N access points (APs) shown in Fig. 4(b) at a height of 2.2m on the office floor. Meanwhile, we collected received signal strength (RSS) samples with a Meizu m2 smart phone at a height of 1.2m. We installed a self-developed Android application on the smart phone and the RSS samples could be collected by the smart phone at a rate of one RSS sample per second. In Room 620, a total of 1920 RSS samples were collected at 16 reference points (RPs) to establish a radio-map for fingerprinting localization and 780 RSS samples were collected for testing the fingerprinting and PM localization performance.



(a) (b) (c) (d)

FIGURE 5. Results of image processing and pixel location search: (a) resized and rotated image, (b) gray scale image, (c) constructed background image, (d) detected foreground target with searched pixel location.

B. RESULTS OF TARGET DETECTION AND PIXEL LOCATION SEARCH

As mentioned before, we detect a human target and search the target's pixel location in the image. We take one testing image as an example and the results of image processing and pixel location search are shown in Fig. 5. Specifically, for performance improvement, the observed image is first resized and rotated shown in Fig. 5(a). The obtained gray scale image is shown in Fig. 5(b). In Fig. 5(c), we can see the background image that is modeled with (1). Then we calculate the pixel difference values between the current gray scale image and background image using (2). In this paper, we empirically set the threshold T in (2) to be 32 to distinguish whether a pixel

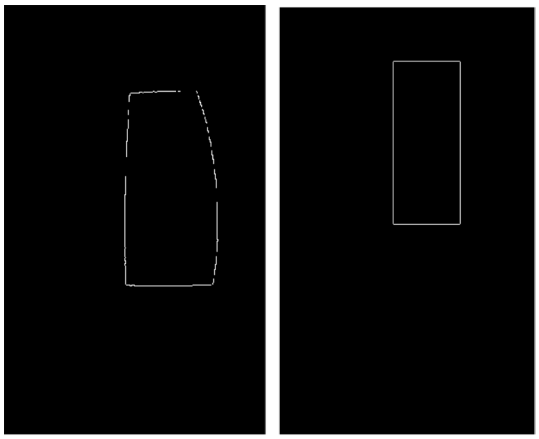


FIGURE 6. Edge detection results of the maximum object: (a) detected edge of the maximum object in the image, (b) detected edge of the maximum object in the layout map.

is a foreground pixel or not. So the foreground target can be determined and shown in Fig. 5(d). According to (3), the minimum Euclidean distance between a foreground pixel location and the pixel location of the panoramic camera is calculated and searched. Then the human target's pixel location in the image can be determined, which is denoted with a red dot in Fig. 5(d).

C. RESULTS OF EDGE DETECTION AND AREA PARTITION

After we obtain the background image shown in Fig. 5(c), we binarize it and apply morphological operations to it for recognizing the connected components accurately. Then we label all the connected components with different digits and the maximum component, that is the maximum object, can be recognized in the observed image plan. We detect the edge of the recognized maximum object with Canny edge detection algorithm. In the same way, we also detect the edge of the maximum object in the layout map. The edge detection results in the image and layout map are shown in Fig. 6.

As shown in Fig. 7, the edge pixel locations in the image and layout map are divided into four sets denoted with different colors. The numbers of the detected edge pixel locations in the up-left, down-left, down-right, and up-right subareas of the image are 281, 115, 172, and 350, respectively. Meanwhile, the numbers of the detected edge pixel locations in the four subareas of the layout map are 281, 84, 177, and 374, respectively. As we can see, only the edge pixel locations in the up-left subareas of the image and layout map are equal in number, so we need the pixel location matching algorithm to make the edge pixel locations of each subarea equal in number and then be matched.

D. RESULTS OF PIXEL LOCATION MATCHING AND TRAINING DATA AUGMENTATION

We utilize the pixel location matching algorithm to match the pixel locations in the image with the ones in the layout map. The result of the matching algorithm is shown in Fig. 8.

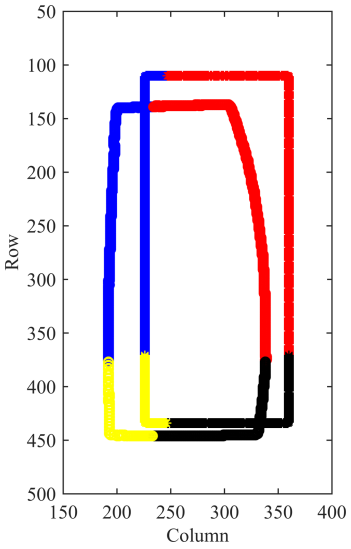


FIGURE 7. Results of area partition for edge pixel locations in the image and layout map.

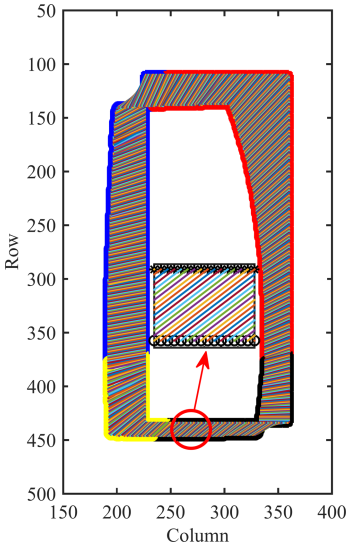


FIGURE 8. Result of pixel location matching algorithm.

We connect the pixel locations in the image denoted by "o" with the pixel locations in the layout map denoted by "*" to show the result clearly. In each subarea, the pixel locations in the image and layout map are equal in number and matched in turn. The numbers of the pixel location pairs in the four subareas after the pixel location matching are 281, 84, 172, and 350, respectively.

We then augment the training data for constructing the regression models according to the symmetry. We take the up-left subarea of the image as an example. The set of edge pixel locations in the up-left subarea is denoted by $\mathbf{O}'_1$ as we mentioned in Section III-D. The transformed pixel location sets from the other three subareas to the up-left subarea can be denoted by $\mathbf{O}'_{2,1}$, $\mathbf{O}'_{3,1}$, and $\mathbf{O}'_{4,1}$. Then there are four sets of the edge pixel locations in the up-left subarea of the image that are $\mathbf{O}'_1$, $\mathbf{O}'_{2,1}$, $\mathbf{O}'_{3,1}$, and $\mathbf{O}'_{4,1}$. They are denoted by blue "*" in Fig. 9. The transformed sets of the pixel

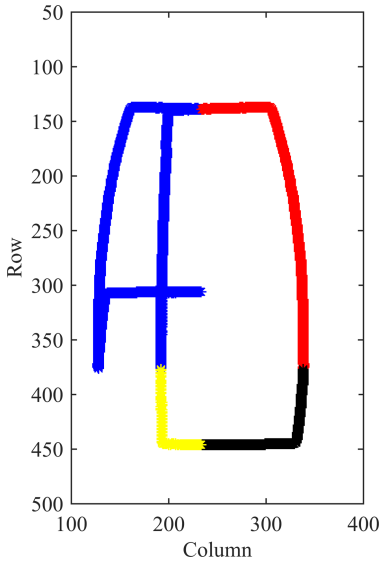


FIGURE 9. Result of training data augmentation in the up-left subarea of the image.

TABLE 1. Performance comparison of various localization methods.

Method	Mean Error (m)	Cumulative Probability (%) Within 1m error	Within 2m error
KNN	1.70	33.6	63.3
WKNN	1.65	37.9	62.6
MLP	1.73	34.4	64.9
SVM	1.56	32.1	68.2
PM	2.33	17.2	53.6
Camera	1.19	32.8	99.9

locations in the up-left subarea of the layout map can be calculated in the same way and the four sets of the edge pixel locations in the up-left subarea of the layout map can be denoted by $\mathbf{L}'_1$, $\mathbf{L}'_{2,1}$, $\mathbf{L}'_{3,1}$, and $\mathbf{L}'_{4,1}$. So the total number of the pixel location pairs in the up-left subarea increases to 887 from 281. We also generate the training data for the other three subareas and the number of pixel location pairs in each subarea increases to 887. The 887 pixel locations in the other three subareas have the same relative location relationship as the 887 pixel locations in the up-left subarea. So we just show the 887 pixel locations in the up-left subarea of the image. Then the total number of the pixel location pairs in the whole target area increases to $887 \times 4 = 3548$, which is also the total number of the generated training data samples.

E. LOCALIZATION RESULTS AND ANALYSES

We use 1200 images as the testing data that are derived from the videos recorded by the panoramic camera. We first use 887 basic training data samples without training data augmentation to construct a GRNN model for the whole target area. For performance comparison, we perform the K nearest neighbors (KNN), weighted K nearest neighbors (WKNN), MLP, and support vector machine (SVM) fingerprinting localization algorithms as well as PM localization method with optimized parameters in Room 620. The experimental results are listed in Table 1. The mean errors of the KNN, WKNN, MLP, and SVM fingerprinting localization algorithms are 1.70m, 1.65m, 1.73m, and 1.56m, respectively. The mean error of the PM localization method with optimized parameters is 2.33m. Meanwhile, the mean error of the proposed panoramic camera-based localization using the GRNN model without training data augmentation is 1.19m, which is less than those of the other localization methods.

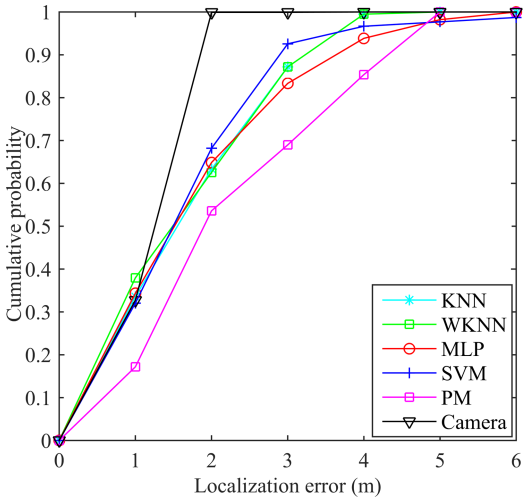


FIGURE 10. Cumulative probabilities of various localization methods.

TABLE 2. Performance comparison of panoramic camera-based localization with training data augmentation and different regression models.

Model	Mean Error (m)	Cumulative Probability (%) Within 1m error	Within 2m error
MLP	0.92	69.6	92.2
ANFIS	1.10	64.2	89.3
RBF	1.00	60.7	92.1
SVM	0.83	69.9	99.6
ML	0.86	60.3	99.8
GRNN	0.77	85.9	91.7

The cumulative probability curves of these localization methods are shown in Fig. 10. The cumulative probabilities of the KNN, WKNN, MLP, SVM, PM, and panoramic camera-based localization within a localization error of 1m are 33.6%, 37.9%, 34.4%, 32.1%, 17.2%, and 32.8%, respectively. The cumulative probability of panoramic camera-based localization method within a localization error of 2m is much higher than those of the other localization methods. In Fig. 10, almost all the localization errors of the panoramic camera-based localization method are less than 2m and its localization performance outperforms the other methods.

With the training data augmentation, we have 887 training data samples in each subarea and 3548 training data samples in the whole target area. Besides the GRNN, we also use the MLP, adaptive neural fuzzy inference system (ANFIS), RBF, SVM, and maximum likelihood (ML)-based regression models. As shown in Table 2, the mean errors of the MLP, ANFIS, RBF, SVM, ML, and GRNN are 0.92m, 1.10m, 1.00m, 0.83m, 0.86m, and 0.77m, respectively. Our proposed localization method using GRNN has a superior performance.

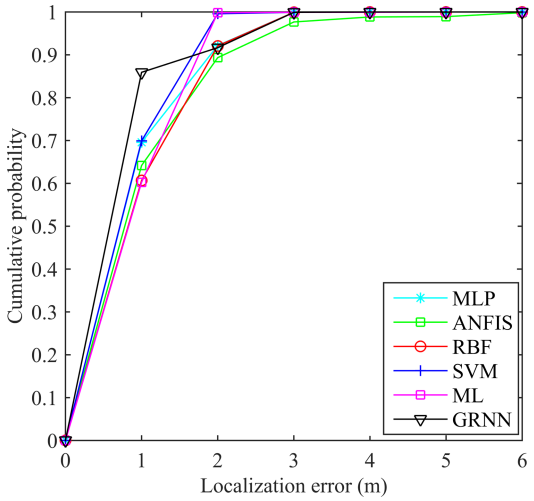


FIGURE 11. Cumulative probabilities of panoramic camera-based localization with training data augmentation and different regression models.

Compared with our previous panoramic camera-based localization method using MLP in [5], whose mean error is 0.84m, although the mean error of the localization method using GRNN without training data augmentation in this paper is 1.19m, it is able to generate the training data automatically, which saves time and labor costs for collecting the training data, let alone the mean error of the proposed panoramic camera-based localization method can be reduced to 0.77m with the training data augmentation in this paper.

The cumulative probability curves using these different regression models are shown in Fig. 11. The cumulative probabilities of the MLP, ANFIS, RBF, SVM, ML, and GRNN within a localization error of 1m are 69.6%, 64.2%, 60.7%, 69.9%, 60.3%, and 85.9%, respectively. The cumulative probabilities of the MLP, ANFIS, RBF, SVM, ML, and GRNN within a localization error of 2m are 92.2%, 89.3%, 92.1%, 99.6%, 99.8%, and 91.7%, respectively. Within a localization error of 1m, the cumulative probability of the GRNN is much higher than those of the other regression models. Within a localization error of 2m, the cumulative probability of the GRNN is a little lower than those of the MLP, RBF, SVM, and ML. We can know that more localization errors greater than 2m are computed by the GRNN. In conclusion, the GRNN generally outperforms the other regression models.

V. CONCLUSION AND FUTURE WORKS

In this paper, we propose a panoramic camera-based human localization method using automatically generated training data. Using only one panoramic camera to monitor a room, the proposed method is able to not only locate a human target accurately without any terminal device, but also generate training data automatically for constructing the GRNN models. We first recognize a feature object and detect the edge pixels of the object in the image and layout map. Then we partition the target area into four subareas and match the edge pixel locations of each subarea in the image with the ones in the layout map to generate training data. We also augment the training data using the symmetry and construct one GRNN model for each subarea. When a human target is detected, the human target's pixel location in the image can be determined. The target's pixel location coordinates along with the radius and angle are input into one of four GRNN models to calculate the target's location coordinates in the layout map, which are transformed to be the localization coordinates in the coordinate system we constructed. Experimental results verify that our proposed panoramic camera-based human localization method using automatically generated training data outperforms the fingerprinting and PM localization methods. In practical applications, although the proposed panoramic camera-based localization method is not suitable for all the application scenarios, it might be an effective solution for some localization application scenarios in room environments. Meanwhile, it could also be integrated with other localization methods for human localization in indoor environments.

In the future, we might extend the target area through fusing the observed images from multiple panoramic cameras, improve the localization performance by using advanced target detection methods, and integrate human action recognition into the proposed localization method.

REFERENCES

[1] L. Chen, S. Thombre, K. Jarvinen, E. S. Lohan, A. Alen-Savikko, H. Leppakoski, M. Z. H. Bhuiyan, S. Bu-Pasha, G. N. Ferrara, S. Honkala, J. Lindqvist, L. Ruotsalainen, P. Korpisaari, and H. Kuusniemi, "Robustness, security and privacy in location-based services for future IoT: A survey," *IEEE Access*, vol. 5, pp. 8956–8977, 2017.

[2] Y. Gu, A. Lo, and I. Niemegeers, "A survey of indoor positioning systems for wireless personal networks," *IEEE Commun. Surveys Tuts.*, vol. 11, no. 1, pp. 13–32, 1st Quart., 2009.

[3] C. Laoudias, A. Moreira, S. Kim, S. Lee, L. Wirola, and C. Fischione, "A survey of enabling technologies for network localization, tracking, and navigation," *IEEE Commun. Surveys Tuts.*, vol. 20, no. 4, pp. 3607–3644, 4th Quart., 2018.

[4] M. Zhou, Y. Tang, Z. Tian, and X. Geng, "Semi-supervised learning for indoor hybrid fingerprint database calibration with low effort," *IEEE Access*, vol. 5, pp. 4388–4400, 2017.

[5] Y. Sun, W. Meng, C. Li, N. Zhao, K. Zhao, and N. Zhang, "Human localization using multi-source heterogeneous data in indoor environments," *IEEE Access*, vol. 5, pp. 812–822, 2017.

[6] D. Zou, W. Meng, S. Han, K. He, and Z. Zhang, "Toward ubiquitous LBS: Multi-radio localization and seamless positioning," *IEEE Wireless Commun.*, vol. 23, no. 6, pp. 107–113, Dec. 2016.

[7] F. Xiao, Z. Wang, N. Ye, R. Wang, and X.-Y. Li, "One more tag enables fine-grained RFID localization and tracking," *IEEE/ACM Trans. Netw.*, vol. 26, no. 1, pp. 161–174, Feb. 2018.

[8] L. Song, T. Zhang, X. Yu, C. Qin, and Q. Zhang, "Scheduling in cooperative UWB localization networks using round trip measurements," *IEEE Commun. Lett.*, vol. 20, no. 7, pp. 1409–1412, Jul. 2016.

[9] Y. Sun, X. Zhang, X. Wang, and X. Zhang, "Device-free wireless localization using artificial neural networks in wireless sensor networks," *Wireless Commun. Mobile Comput.*, vol. 2018, Jun. 2018, Art. no. 4201367.

[10] S.-H. Fang and T.-N. Lin, "Cooperative multi-radio localization in heterogeneous wireless networks," *IEEE Trans. Wireless Commun.*, vol. 9, no. 5, pp. 1547–1551, May 2010.

[11] Z. Deng, W. Si, Z. Qu, X. Liu, and Z. Na, "Heading estimation fusing inertial sensors and landmarks for indoor navigation using a smartphone in the pocket," *EURASIP J. Wireless Commun. Netw.*, vol. 2017, no. 1, p. 160, Dec. 2017.

[12] W. Zhao, S. Han, R. Q. Hu, W. Meng, and Z. Jia, "Crowdsourcing and multisource fusion-based fingerprint sensing in smartphone localization," *IEEE Sensors J.*, vol. 18, no. 8, pp. 3236–3247, Apr. 2018.
[13] M. Idoudi, E.-B. Bourennane, and K. Grayaa, "Wireless visual sensor network platform for indoor localization and tracking of a patient for rehabilitation task," *IEEE Sensors J.*, vol. 18, no. 14, pp. 5915–5928, Jul. 2018.
[14] E. Kougianos, S. P. Mohanty, G. Coelho, U. Albalawi, and P. Sundaravadivel, "Design of a high-performance system for secure image communication in the Internet of Things," *IEEE Access*, vol. 4, pp. 1222–1242, 2016.
[15] L. Liu, X. Zhang, and H. Ma, "Optimal node selection for target localization in wireless camera sensor networks," *IEEE Trans. Veh. Technol.*, vol. 59, no. 7, pp. 3562–3576, Sep. 2010.
[16] L. Liu, X. Zhang, and H. Ma, "Localization-oriented coverage in wireless camera sensor networks," *IEEE Trans. Wireless Commun.*, vol. 10, no. 2, pp. 484–494, Feb. 2011.
[17] E. Lobaton, R. Vasudevan, R. Bajcsy, and S. Sastry, "A distributed topological camera network representation for tracking applications," *IEEE Trans. Image Process.*, vol. 19, no. 10, pp. 2516–2529, Oct. 2010.
[18] Á. Utasi and C. Benedek, "A Bayesian approach on people localization in multicamera systems," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 23, no. 1, pp. 105–115, Jan. 2013.
[19] Y. Liu, Q. Wang, J. Liu, J. Chen, and T. Wark, "An efficient and effective localization method for networked disjoint top-view cameras," *IEEE Trans. Instrum. Meas.*, vol. 62, no. 9, pp. 2526–2537, Sep. 2013.
[20] Y.-S. Lin, K.-H. Lo, H.-T. Chen, and J.-H. Chuang, "Vanishing point-based image transforms for enhancement of probabilistic occupancy map-based people localization," *IEEE Trans. Image Process.*, vol. 23, no. 12, pp. 5586–5598, Dec. 2014.
[21] M.-C. Lu, C.-C. Hsu, and Y.-Y. Lu, "Image-based system for measuring objects on an oblique plane and its applications in 2-D localization," *IEEE Sensors J.*, vol. 12, no. 6, pp. 2249–2261, Jun. 2012.
[22] J. Miseikis and P. V. K. Borges, "Joint human detection from static and mobile cameras," *IEEE Trans. Intell. Transp. Syst.*, vol. 16, no. 2, pp. 1018–1029, Apr. 2015.
[23] S. Minaeian, J. Liu, and Y.-J. Son, "Vision-based target detection and localization via a team of cooperative UAV and UGVs," *IEEE Trans. Syst., Man, Cybern., Syst.*, vol. 46, no. 7, pp. 1005–1016, Jul. 2016.
[24] M. Nixon, *Feature Extraction and Image Processing for Computer Vision*, 3rd ed. New York, NY, USA: Academic, 2012, pp. 435–450.
[25] J. Canny, "A computational approach to edge detection," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. PAMI-8, no. 6, pp. 679–698, Nov. 1986.
[26] R. Haralick and L. Shapiro, *Computer and Robot Vision*, vol. 1. Reading, MA, USA: Addison-Wesley, 1992, pp. 28–48.
[27] D. F. Specht, "A general regression neural network," *IEEE Trans. Neural Netw.*, vol. 2, no. 6, pp. 568–576, Nov. 1991.
[28] D. Tomandl and A. Schober, "A modified general regression neural network (MGRNN) with new, efficient training algorithms as a robust 'black box'-tool for data analysis," *Neural Netw.*, vol. 14, no. 8, pp. 1023–1034, Oct. 2001.

YONGLIANG SUN (Member, IEEE) received the B.Eng. degree in electronic information engineering from Harbin Engineering University, in 2006, and the M.Eng. and Ph.D. degrees in information and communication engineering from the Harbin Institute of Technology (HIT), in 2008 and 2014, respectively. He is currently a Lecturer with the School of Computer Science and Technology, Nanjing Tech University. His research interests include indoor localization, communications networks, and the Internet of Things.

WEIXIAO MENG (Senior Member, IEEE) received the B.Eng., M.Eng., and Ph.D. degrees from the Harbin Institute of Technology (HIT), Harbin, China, in 1990, 1995, and 2000, respectively. From 1998 to 1999, he worked as a Senior Visiting Researcher at NTT DoCoMo on adaptive array antennas and dynamic resource allocation for beyond 3G. He is currently a Full Professor and the Vice Dean of the School of Electronics and Information Engineering, HIT. His research interests include broadband wireless communications and networking, MIMO, and indoor localization technologies. He has published three books and over 220 articles in journals and international conferences. He is a Fellow of the China Institute of Electronics and a Senior Member of the China Institute of Communication. He is also the Chair of the IEEE Communications Society Harbin Chapter.

CHENG LI (Senior Member, IEEE) received the B.Eng. and M.Eng. degrees from the Harbin Institute of Technology, Harbin, China, in 1992 and 1995, respectively, and the Ph.D. degree in electrical and computer engineering from Memorial University, St. Johns, NL, Canada, in 2004. He is currently a Full Professor with the Department of Electrical and Computer Engineering, Faculty of Engineering and Applied Science, Memorial University. His research interests include mobile ad hoc and wireless sensor networks, wireless communications and mobile computing, switching and routing, and broadband communication networks.

XUZI WU received the B.Eng. degree in information countermeasure technology and the Ph.D. degree in pattern recognition and intelligent system from Xidian University, in 2011 and 2017, respectively. She is currently a Lecturer with the School of Computer Science and Technology, Nanjing Tech University. Her research interests include array signal processing, radar signal processing, and digital image processing.