Hindawi Publishing Corporation
Journal of Sensors
Volume 2015, Article ID 760435, 12 pages
http://dx.doi.org/10.1155/2015/760435

Research Article

Person Tracking System by Fusing Multicues Based on Patches

Song Min Jia,[1,2,3] Li Jia Wang,[1,2,4] Xiu Zhi Li,[1] and Lin Feng Wen[1]

[1] *College of Electronic Information & Control Engineering, Beijing University of Technology, No. 100, Pingleyuan, Chaoyang District, Beijing 100124, China*
[2] *Beijing Key Laboratory of Computational Intelligence and Intelligent System, Beijing 100124, China*
[3] *Engineering Research Center of Digital Community, Ministry of Education, Beijing 100124, China*
[4] *Department of Information Engineering and Automation, Hebei College of Industry and Technology, No. 626, Hongqi Street, Qiaoxi District, Shijiazhuang 050081, China*

Correspondence should be addressed to Song Min Jia; jsm@bjut.edu.cn

Received 15 January 2015; Revised 13 May 2015; Accepted 20 May 2015

Academic Editor: Yasuko Y. Maruo

A person tracking algorithm by fusing multicues based on patches is proposed to solve the problem of distinguishing person, occlusion, and illumination variations. Kinect is mounted on the robot for providing color images and depth maps. A detector representing a person by using the fusion of multicues based on patches is proposed. The detector divides the person into many patches and then represents each patch by using depth-color histograms and depth-texture histograms. The appearance representation, considering depth, color, and texture information, has powerful discrimination ability to handle the problems of occlusion, illumination changes, and pose variations. Considering the motion of the robot and person, a tracker called motion extended Kalman filter (MEKF) is presented to predict the person's position. The result of the tracker is treated as a candidate sample of the detector, and then the result of the detector is the previous knowledge of the tracker. The detector and tracker supplement each other and improve the tracking performance. To drive the robot towards the given person precisely, a fuzzy based intelligent gear control strategy (FZ-IGS) is implemented. Experiments demonstrate that the proposed approach can track a person in a complex environment and have an optimum performance.

1. Introduction

With the popularity of robot in human environments, it is necessary to detect and track a person in many applications including surveillance, search, rescue, combat, and human assistant. Person detecting and tracking are very challenging computer vision tasks due to automatic initialization, pose variations, expensive calculation cost, and occlusions in complicated environments [1].

In real-world settings, persons are nonrigid and difficult to be tracked. To resolve the problem, an efficient representation should be considered for an available appearance model. Color is widely used for modeling a target, and one of the best methods for color-based object tracking is to realize the mean shift algorithm [2, 3]. Ning et al. [4] presented a scale and orientation adaptive mean shift algorithm to handle the problem of scale and orientation changes. Unfortunately, the pixel-wise color density does not consider extreme geometric changes of an object. It is vulnerable when there is occlusion or similar background. Many researches have focused on resolving the problem by utilizing the texture feature. Ning et al. [5] used joint color-texture histograms for robustly tracking a target in complex environment. Compared with the traditional color histogram, the joint color-texture histograms efficiently exploited a target's structure information and hence performed better when a target has similar color appearance with the background.

To further eliminate the influence of background, depth information captured from stereo cameras is employed. The depth information easily performs the foreground-background segmentation [6]. However, most stereo tracking systems are implemented with known calibration parameters [7]. In the last few years, stereocameras (e.g., Kinect [8]), with no extensive knowledge of camera calibration parameters and low cost, have been widely used in computer vision [9, 10]. Compared with the traditional stereocameras [7], Kinect can provide higher quality color image and depth map and is widely employed recently. Xia et al. [10] considered object tracking as foreground-background segmentation by extracting contour information and depth feature from a Kinect sensor. Zoidi et al. [6] represented an appearance by fusing Local Steering Kernel features and 2D color-disparity histograms. The method employed disparity information to identify scale changes by analyzing disparity values. The depth image (disparity image) indicates objects' distances in the complexity environment, which meets the human visual perception system. Therefore, the depth information is of great significance to discriminate the target from the background.

Occlusion is a difficult problem in object tracking. To cope with the problem, patches based algorithms were proposed [11–14]. Adam et al. [11] presented a fragments-based color histograms method. The method represented a target by integrating each part's color histograms to handle partial occlusions and pose variations. Nejhum et al. [12] used multiple blocks to model a frequently changing foreground shape. The method successfully tracked objects undergoing significant shape variations and illumination changes. Yang et al. [13] proposed a spatially attentional patches based tracking method which performed well on a large number of real-world videos. Kwon and Lee [14] proposed a patches based dynamic appearance model for representing a target. The hue, saturation, and value features were adaptively selected for calculating photometric likelihood, while the squared differences between patches were adopted for representing geometric likelihood. Unfortunately, it suffered from a high computing burden due to the Basin Hopping Monte Carlo.

For a robot system in clustered environment, a continuous and stable controller is important for following a person. However, to the author's knowledge, many works have focused on the problem of target detection and tracking but rarely addressed the problem of designing a suitable controller for driving a robot [15]. The existing controller mainly includes the PID controller [16], visual based sliding mode controller [15], fuzzy based controller [17], and intelligent controller [7]. Ouadah et al. [15] presented two sliding mode controllers to control the robot according to the person's position obtained from RFID system and visual system, respectively. The robot can follow the given person when there is a sudden turn. Jia et al. [7] presented an intelligent speed controller considering the robot's kinematics. However, the algorithm often fails to follow the person because of the fixed linear velocity.

In the past decades, person tracking system using a robot has achieved a lot of improvements. However, the problems of distinguishing person, occlusion, and safe following still exist. We address the problem by representing a person with multicues based on patches and designing a fuzzy based intelligent gear control strategy (FZ-IGS). The person detection algorithm includes a detector and a tracker. The detector divides a person into many patches and represents a patch by the use of multicues including depth, color, and texture. The depth information, indicating the person's location, is combined with color and texture features for generating depth-color histograms and depth-texture histograms, respectively. As track evolves, the detector adjusts the person's size according to depth information. By analyzing the depth histograms and patches' similarity with the given person, the detector can easily recognize the occlusion and then make a decision to update the person's appearance model and change the tracking strategy. When there is a partial occlusion, the detector recognizes the person by using the patches which are not occluded. The tractor called MEKF is generated from the EKF by considering the motion of the robot and person. The MEKF predicts the person's position as a candidate sample for the detector. Finally, FZ-IGS is designed to change the turning gain and linear velocity of the robot according to the position of the person from the robot. The FZ-IGS drives the robot towards the person continuously and stably.

The paper is organized as follows: the overview of the proposed method is discussed in Section 2. In Section 3, the multicues based detector is described. Section 4 details the steps of processing person location and model update. The fuzzy based controllers are described in Section 5. The experimental results are detailed in Section 6. The paper conclusion with a short summary is shown in Section 7.

2. Framework and Architecture

The section details the platform and the system overviews for performing the person following task.

2.1. Development Platform and Environment. The platform used for performing person following task is an American Mobile Robots Inc. Pioneer 3-DX embedded with a Kinect, illustrated in Figure 1. The Kinect is a new and widely available device for the Xbox 360. The interest for Kinect is increasing in computer vision due to its advantages of providing 3D information of the environment and low cost. The device contains an RGB camera, a multiarray microphone, and a depth sensor. Using these sensors, Kinect can capture full body 3D motion. The Kinect hardware specification is detailed as follows:

(1) RGB camera: 640 × 480 pixels/32 bit colour at 30 frames/sec,

(2) depth sensor: 320 × 240 pixels/16 bit greyscale at 30 frames/sec,

(3) sensor range: 1.2 m–3.5 m,

(4) field of view: horizontal: 57° (1.3 m–3.8 m); vertical: 43°.

Using these sensors, the Kinect can provide two kinds of images: depth image and color image. The depth image is obtained by the depth sensor which contains a CMOS camera and an infrared projector. The infrared projector produces speckle pattern in the scene. Then, the CMOS camera records the speckle pattern and results in the depth image. The color image is produced by the RGB camera with a resolution of 640 × 480 pixels at 30 frames per second. The Kinect has a

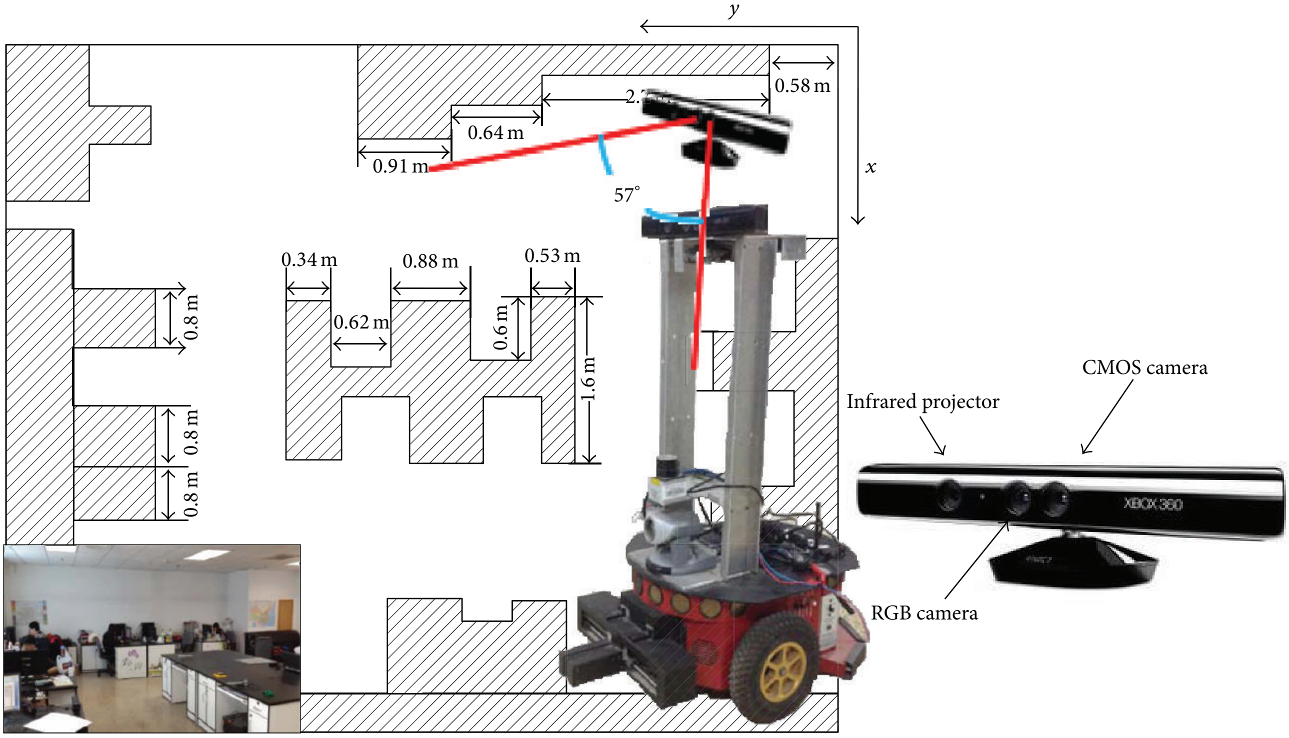


Figure 1: The platform for person tracking.

field of view 57° which can satisfy the need of object tracking. The algorithm is implemented by VC++2008 and Opencv2.1.

2.2. System Overview. Given a stream of color images and depth maps, our goal is to continuously track a person. The overview of our system is presented in Figure 2. The system includes a detector, a tracker, and an online update strategy. The detector represents the person by using the depth, color, and texture information obtained from the Kinect. The tracker predicts the person's position by considering the person and robot's motion. The result of the tracker is treated as a candidate sample of the detector for determining the person's location, and then the result of the detector is adopted as the previous information of the tracker. The detector and tracker supplement and complement each other, which improves the tracking performance. As track evolves, the detector adjusts the person's size according to depth histograms and determines the occlusion problem based on the depth histograms and patches' appearance similarity. Finally, the online update strategy adaptively updates the person's appearance to avoid introducing more inference and handle the variations on illumination and pose.

3. Detector

It is reported that the appearance represented by a single feature often fails in tracking process when there is similar background. To handle the problem, we represent a person by using multicues including depth, color, and texture. The detector can successfully recognize a person by using one feature while the other features are invalid. The depth feature, easily discriminating the person from background, is extracted for representing the person to overcome the background's inference. Furthermore, the detector detects the problem of occlusion considering the depth histograms and the patches' appearance similarity and then adjusts the online update strategy.

3.1. Depth Histograms. Depth map, captured from the sensor Kinect, provides 3D information of the environment and is invariant to illumination [6]. Compared with the color image, the depth values provide an intuitive notion of the relative person's distance from the robot. The larger the depth value is, the closer the person is to the robot. In our case, the robot is controlled towards the given person and remains in a safety distance from him. Therefore, the depth values of the person are closest to the robot. Then, the depth segmentation can be performed on the depth map for distinguishing the person from the background.

The depth is discretely distributed in n intervals. The depth values are represented by a vector $x^*_{i\ i=1,2,\ldots,M}$; M is the number of the depth value. A delta function $\delta(\cdot)$ is employed to determine the interval for the depth value x_i. Then, depth features, called depth histograms, are extracted by analyzing the pixels of the depth image:

$$\widehat{q}_d = \sum_{i=1}^{M} \delta\left[b\left(x^*_i\right) - u\right], \tag{1}$$

where u is an interval. During tracking, we assume that the person is in front of the robot and his position from the robot

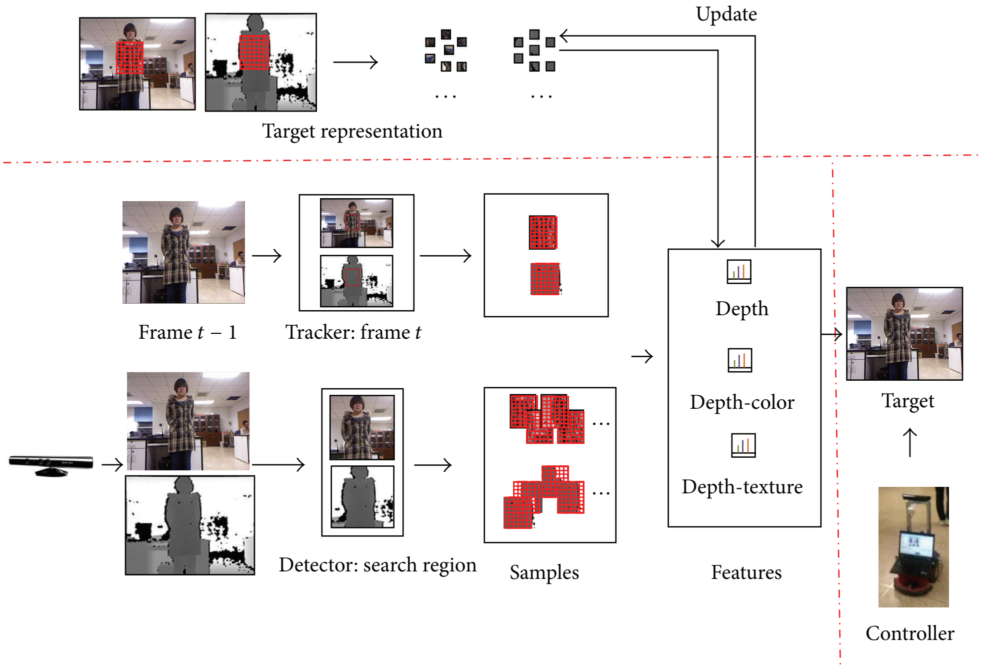


Figure 2: The overview of the system.

does not change significantly from frame to frame. Therefore, all of the person's depth values will lie in the last bin of the depth histograms and will be far from the background. Considering the depth histograms, the foreground-background segmentation will be much easier. The depth histogram is shown in Figure 3. Affected by the illumination, the depth value in some region is higher than another region, shown in Figure 3(a). Therefore, the bins belonging to the target are the last two bins (181 and 211) shown in Figure 3(b). The depth histogram for the target in the blue rectangle is shown in Figure 3(c). It has two bins: bin 181 is for the region with lower depth value; bin 211 is for the region with higher depth value.

Furthermore, the person's size changes according to the variation of his position from the robot in the tracking process. The appearance model obtained by using the fixed rectangle size will introduce background's inference or lose some important information when the distance changes. While the distance is large, we expect the rectangle size to be small for fitting the person. When the person is close to the robot, we expect the rectangle size to be large to fit the person. The depth information indicates the changes of the person's size. In the case in which the person's position changes, the bin values of the depth histograms will correspondingly vary. Thus, we adaptively adjust the rectangle's size based on the depth histograms. The person's current size is obtained as follows:

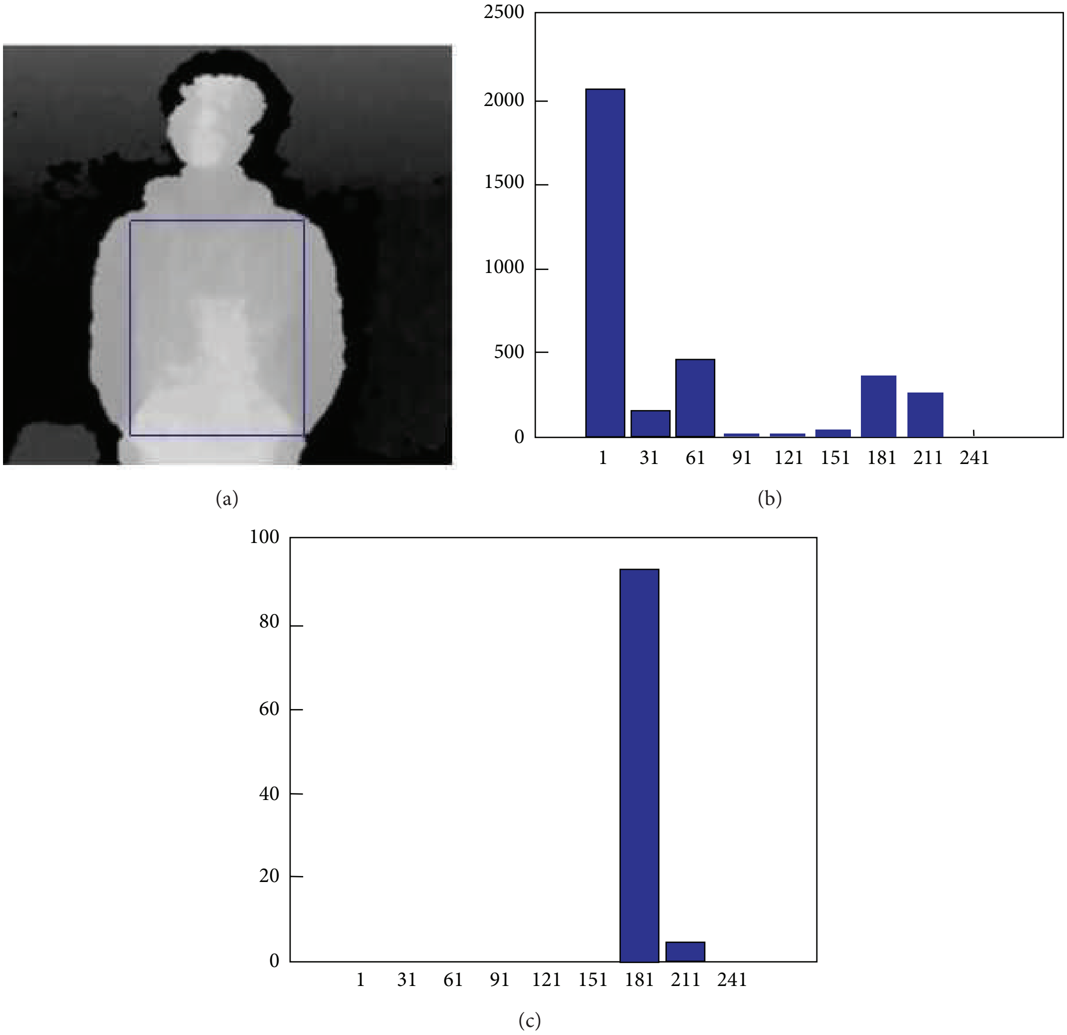
$$\text{size}_{\text{new}} = \text{size}_{\text{old}} \times \gamma, \tag{2}$$

where $\gamma = \widehat{q}_d/\widehat{q}_{\text{base}}$ and $\widehat{q}_d$ is the target's depth histogram. $\widehat{q}_{\text{base}}$ is the reference depth feature which is determined by the initial object. c_{base} is the adjustment parameter which is determined by initial size of the target. $\text{size}_{\text{old}} = \{W, H\}$ is the size of the person in previous frame. When the person's size changes due to the variations of the distance between the person and the robot, the rectangle size is updated. The obtained person's rectangle size, fitting the variations of the distance, can not only avoid inducing more inference from background due to a larger rectangle but also avoid losing important information because of the smaller rectangle. After adjusting the person size, the detector collects the candidate samples based on the new size and updates the appearance model.

3.2. Depth-Color/Texture Detector

3.2. Depth-Color/Texture Detector. In order to successfully discriminate a given person, multicues are employed for representing the person. Color has been proved to be useful for modeling a target. Compared with other features, color is insensitive to scale and translation. Therefore, it has been widely adopted for target representation. Texture, as another effective description operator, indicates the pixels' space property. To obtain more powerful representation, color and texture are mixed for modeling a target.

The traditional color and texture based object representation has successfully discriminated person when there is color or texture clutter in background [5]. However, it can hardly solve the problem of occlusion and complex background. In our research, depth information is employed to handle color or texture clutter in background due to the depth's ability of foreground-background segmentation. The disadvantage of depth segmentation is that it cannot easily discriminate the objects lying in the same distance from the robot. Fortunately, this can be resolved by using the color or texture features. The depth, color, and texture are combined to generate the depth-color histograms and depth-texture histograms for representing the person. Moreover, to deal with the occlusion problem, patches based representation method is presented.


(a)

(b)

(c)

Figure 3: The illustration of the depth histogram. (a) The depth image, the blue rectangle is for the target. (b) The depth histogram for the depth image. (c) The depth histogram for the target in the blue rectangle.

The person in a rectangle is divided into $N \times N$ patches and each patch is represented by the depth-color and depth-texture histograms:

$$\widehat{q}_{f,n,u} = C\sum_{i=1}^{M} K\left(y_0, x_i^*\right)\delta\left[b\left(x_i^*\right) - u\right], \quad (3)$$

where $f = Cd, Ld$ indicates the depth-color and depth-texture information, respectively. $\widehat{q}_f$ is the obtained depth-color histograms and depth-texture histograms. The color feature is captured from the HSV space, while the texture information is the uniform texture [5]. $n = N \times N$ is the number of the person's patches. $C = 1/\sum_{i=1}^{M} K(y_0, x_i^*)$ is the normalized coefficient. $\{x_i^*\}_{\{i=1,\ldots,M\}}$ is the pixels of each patch; y_0 is the center of each patch. $\delta(\cdot)$ is the *delta* function for determining the feature's bin number. $K(y_0, x_i^*)$ is a kernel function which affects the obtained features' discriminative power. The Epanechnikov function is commonly used. It assigns a larger weight for the pixels in the center of the target image and a smaller value for the pixels far away from the center. This method can avoid introducing to a certain extent the inference of the background around the person. However, for the pixels far away from the center, its importance in the appearance representation is reduced due to the smaller weight. Furthermore, the edges and background far away from the center of an irregular target (e.g., person) may be confused. In such a case, the pixels in the background with smaller weights are introduced into the appearance model. To deal with the problem, depth information is used for constructing the new kernel function for segmenting the target from the background:

$$K\left(y_0, x_i^*\right) = M_{\text{depth},n}\left(x_i^*\right), \quad (4)$$

where

$$M_{\text{depth},n}\left(x_i^*\right) = \begin{cases} g\left(u^*\right), & x_i^* \in \widehat{q}_d\left(u^*\right), \\ 0, & \text{otherwise} \end{cases} \quad (5)$$

is the mask image. u^* is the bin belonging to the person. The new kernel function avoids introducing the background's inference because the pixels with the value $g(u^*)$ in the mask

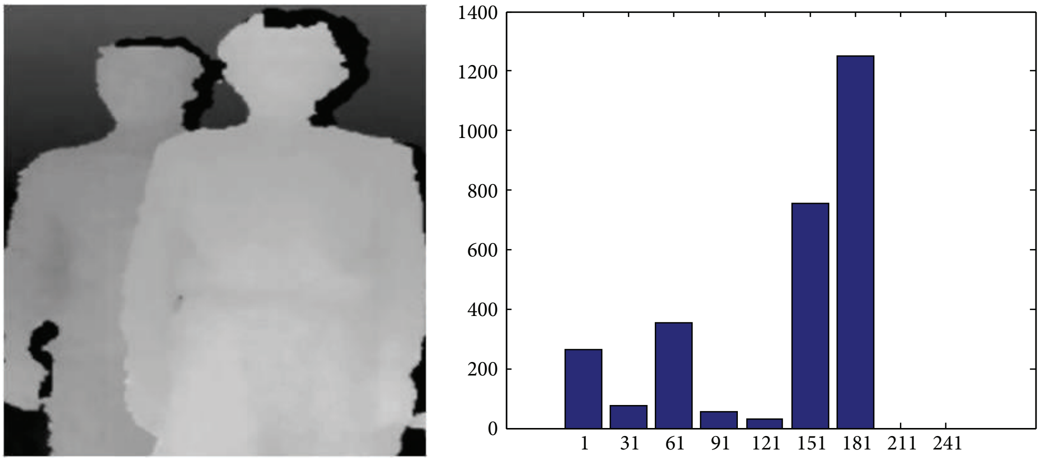


Figure 4: The illustration of depth information when there is occlusion.

image belong to the person and the pixels with the value 0 belong to the background. Compared to the Epanechnikov function, the new function assigns the pixels of the person the same weights to improve the representation's discrimination ability.

The person is represented by modeling each patch using the obtained depth-color and depth-texture histograms. The color histograms describe the target integrally, while the texture histograms depict image's local texture. The two features somehow supplement each other. The depth information, identifying the target from background, deals with the color or texture clutter in background.

3.3. Occlusion Problem.

3.3. Occlusion Problem. As tracking evolves, there may be occlusion which will result in tracking failure. The patches based tracking algorithm was proposed to deal with the problem [18]. The person is divided into $N\times N$ patches. When there is partial occlusion, some patches are occluded and others are free. We present a method to detect the occlusion problem by using the patches based appearance similarity and the depth histograms. After detecting the occlusion, the person is discriminated by processing the unoccluded patches. The depth map with occlusion problem is shown in Figure 4.

For a person tracking system, the person's depth information lies in the last bin of the depth histograms and is far away from the background usually. In the case in which the person is occluded, the last two bins are next to each other. The last bin belongs to the passerby, while the last bin but one is for the person. As shown in Figure 4, the last bin "181" with more than 1200 depth values belongs to the passerby that is near the camera. The last bin but one "151" is for the person that is occluded by the passerby. Then, for the person and the passerby, we perform depth feature and the patches' similarity for detecting the occlusion problem.

In an ideal tracking process, the bin for the person maintains stability. The depth feature similarity is calculated as $S_q = \widehat{q}_d - \widehat{q}_d^t$, where $\widehat{q}_d^t$ is the depth feature in the current frame. A threshold is set to determine whether the depth feature belongs to the target. Moreover, the patches' similarity is processed to detect the person successfully, which will be shown in Section 3.4.

3.4. Patches Based Multicues for Person Detection. Compared with only one feature, to represent a person by extracting different features can improve the model's discrimination ability. Once one feature fails to discriminate the person, the other features are valid. The person is represented by many patches which are in a decreasing order based on the depth-color histograms and depth-texture histograms, respectively. For a given threshold, th, the detector recognizes the person according to their appearance similarity. Normally, the candidate sample with the maximum overall similarity and over 90% of the number of patches is the person. However, some features (e.g., depth-color histograms) may change much due to pose variations or illumination changes. Then, the corresponding similarity will decrease and the overall similarity will be less than the threshold, th. In such a case, the detector will recognize the person based on the other feature's similarity (e.g., depth-texture histograms) and the number of the patches. Once partial occlusion is detected, the detector recognizes the person according to the patches which are not occluded. The patches based multicues representation is shown in Figure 5.

The patch's similarity between the candidate sample and the person is measured by using the cosine similarity metric:

$$\widehat{\rho}_{f,n}\left(\widehat{p}_{f,n},\widehat{q}_{f,n}\right)=\cos(\theta)=\frac{\left\langle \widehat{p}_{f,n},\widehat{q}_{f,n}\right\rangle}{\left\|\widehat{p}_{f,n}\right\|\left\|\widehat{q}_{f,n}\right\|}\in[-1,1], \qquad (6)$$

where $\langle\cdot\rangle$ is the inner product and $\|\cdot\|$ indicates the Euler distance. θ is the angle between two vectors.

The similarity between the candidate sample and the model is

$$\widehat{\rho}_f=\sum_{n=1}^{N\times N}\frac{\widehat{\rho}_{f,n}^2}{1-\widehat{\rho}_{f,n}^2}\in[0,+\infty). \qquad (7)$$

The overall similarity is

$$\widehat{\rho}=\widehat{\rho}_{Cd}\times\widehat{\rho}_{Ld}. \qquad (8)$$

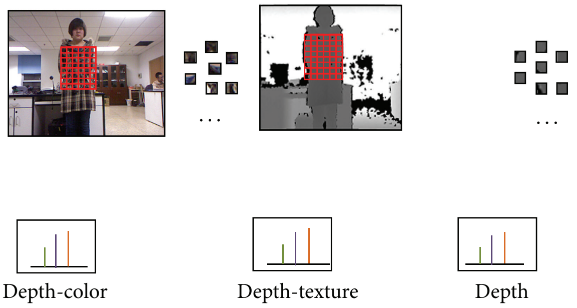


Figure 5: The illustration of depth information when there is occlusion.

3.5. Tracker. EKF is a set of mathematical equations providing an efficient solution for prediction problem. The algorithm is very powerful to deal with the short time occlusion problem in tracking process. However, for the person tracking system with a mobile robot, the EKF often fails to accurately predict the person because the robot and person are moving together. To deal with the problem, we present a tracker called Motion EKF combining the motion of the robot and the person:

$$\begin{aligned} X_r^{t+1} &= f\left(X_r^t, \text{control}_t\right) + R_t w_t, \\ Y_r^t &= H_t X_r^t + p_t, \end{aligned} \tag{9}$$

where $X_r = [x_r, y_r, z_r, \dot{x}_r, \dot{y}_r]$ is the state vector, (x_r, y_r, z_r) is the 3D position of the person in the robot coordinate system, $\dot{x}_r, \dot{y}_r$ are the velocity of the person in the horizontal plane, and $\text{control}_t = [v_l, v_r]$ is the control variable. p_t is the observation noise, and its covariance matrix is $R^t = \text{Cov}(p_t) = E[p_t, p_t^T] = \sigma_p^2 \left[\begin{smallmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{smallmatrix}\right]$. $Y_r^t = (x_r^t, y_r^t, z_r^t)$ is the 3D position of the target in time t. w_t is the process noise, and its covariance matrix is $Q_t = \text{Cov}(w_t) = E[w_t, w_t^T] = \sigma_w^2 \left[\begin{smallmatrix} 1 & 0 \\ 0 & 1 \end{smallmatrix}\right]$. Consider $H_t = \left[\begin{smallmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \end{smallmatrix}\right]$.

Considering the robot's motion, the state transition function is

$$f_t\left(x_r^t, \text{control}_t\right) = \begin{bmatrix} \left(x_r^t + \Delta t x_r^t - \Delta x_r\right)\cos\Delta\theta + \left(y_r^t + \Delta t y_r^t - \Delta y_r\right)\sin\Delta\theta \\ -\left(x_r^t + \Delta t x_r^t - \Delta x_r\right)\sin\Delta\theta + \left(y_r^t + \Delta t y_r^t - \Delta y_r\right)\cos\Delta\theta \\ z_r^t \\ x_r^t \cos\Delta\theta + y_r^t \sin\Delta\theta - v \\ -x_r^t \sin\Delta\theta + y_r^t \cos\Delta\theta \end{bmatrix}. \tag{10}$$

The state equation and observation equation of the MEKF are obtained by considering the robot and person's motion. Compared with EKF, MEKF introduces the robot's trajectories to improve the robustness of the tracking. Moreover, the tracking result is a sample of the candidate set of the detector. The detector recognizes the result from the candidate set including the tracking result. The detector and tracker complement each other, which improves the ability of person detecting.

4. Person Location and Model Update

4.1. Person Location. The proposed tracking framework has been detailed in Figure 2. In the framework, the person is located by applying the detector and the tracker together. During this procedure, the depth information is fused with the color and texture information. Consequently, we obtain depth-color histograms and depth-texture histograms. Then, the person is represented with many patches' appearance models. Furthermore, to realize robustly tracking task, the detector and tracker complement each other. The process for identifying a person is as follows:

(1) Input is as follows: the depth image and color image.

(2) Get depth histograms for the depth image. Divide the depth image and color image into $N \times N$ image patches and then extract these patches' depth-color histograms and depth-texture histograms for modeling the person.

(3) For a new frame, candidate samples are obtained around the result. MEKF predicts the person's position which is treated as a sample for detector. Extract the candidate samples' depth histograms and divide the depth image and color image into $N \times N$.

(4) *For* $n = 1 : N \times N$,

  (a) extract each patch's depth-color histograms and depth-texture histograms;

  (b) compute the patch's similarity of the depth-color histograms and depth-texture histograms.

  End

(5) The patches will be in a decreasing order based on $\hat{\rho}_{Cd,n}$ and $\hat{\rho}_{Ld,n}$. Compute the similarities $\hat{\rho}_{Cd}$ and $\hat{\rho}_{Ld}$. Compute the overall similarity according to $\hat{\rho}$.

(6) Determine the occlusion problem based on the depth histograms and image pieces' similarity, and then detect the person accordingly.

(7) Output is as follows: person's position.

4.2. Model Update. Illumination changes and pose variations may result in appearance variation. To cope with this problem, an efficient update strategy should be used for adjusting to the appearance changes after detecting the person. The update strategy studies the person's appearance model according to patches' similarity in different tracking circumstances:

$$\hat{q}_{f,n,u}^t = \begin{cases} \lambda \times \hat{q}_{f,n,u}^t + (1-\lambda) \times \hat{q}_{f,n,u}^{t-1}, & \lambda > \text{th}, \\ \hat{q}_{f,n,u}^{t-1}, & \text{otherwise}, \end{cases} \tag{11}$$

where $\lambda = \hat{\rho}_{f',n}$ is the patches' appearance similarity.

Normally, the λ is the smaller similarity value of depth-color and depth-texture histograms. When one feature changes too much due to pose or illumination variations and fails to recognize the person, the λ will be determined by the

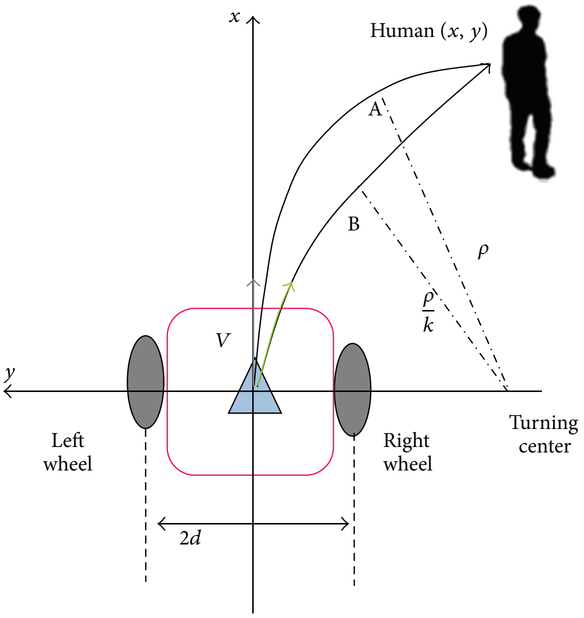


Figure 6: The path of the robot towards the person.

other feature's similarity. In such a case, both of the depth-color and depth-texture histograms are updated based on the λ. In particular, the failure appearance model changes adaptively. When there is partial occlusion, λ equals the unconcluded pieces' similarity.

5. Controller

Our goal is to design an efficient controller to drive the robot towards a given person and remain at a secure distance from him. To follow the robot smoothly and continuously, an intelligence control strategy (IGS) was presented [7], where the robot's speed and steer are controlled through introducing a turning-gain k. Using the turning gain, the robot can adaptively change the turning radius to avoid losing or crashing the person. The path of the robot towards the person is shown in Figure 6. The person's position x_r, y_r, z_r is obtained from the detector mentioned above. ρ is the turning radius of the robot to follow the person.

For path B, the velocities of the robot's wheels are computed as follows:

$$v_l = v\left(1 - \frac{2dky_r}{(x_r^2 + y_r^2)}\right),$$
$$v_r = v\left(1 + \frac{2dky_r}{(x_r^2 + y_r^2)}\right), \quad (12)$$

where v_l and v_r are the velocities of the left wheel and right wheel, respectively. x_r and y_r are the person's positions in the plane coordinate. x_r denotes the direction of the person, while y_r is for his direction.

As following evolves, the turning-gain k and the linear velocity v from the IGS keep constant. For a small turning gain, the turning radius ρ/k is large. When the direction of the person is large (the person is far away from the center of the field of view of the robot), the robot tends to lose the person. In contrast, using a large turning gain, the robot often fails to catch the person close to the center of the robot due to the small turning radius. Similarly, the robot cannot follow the person with large distance by using a small linear velocity and will hit the person due to a large linear velocity. To deal with these problems, we present a fuzzy based intelligent control strategy (FZ-IGS). The strategy includes two fuzzy controllers: a linear velocity controller and a turning-gain controller.

Table 1: The fuzzy logic for velocity controller.

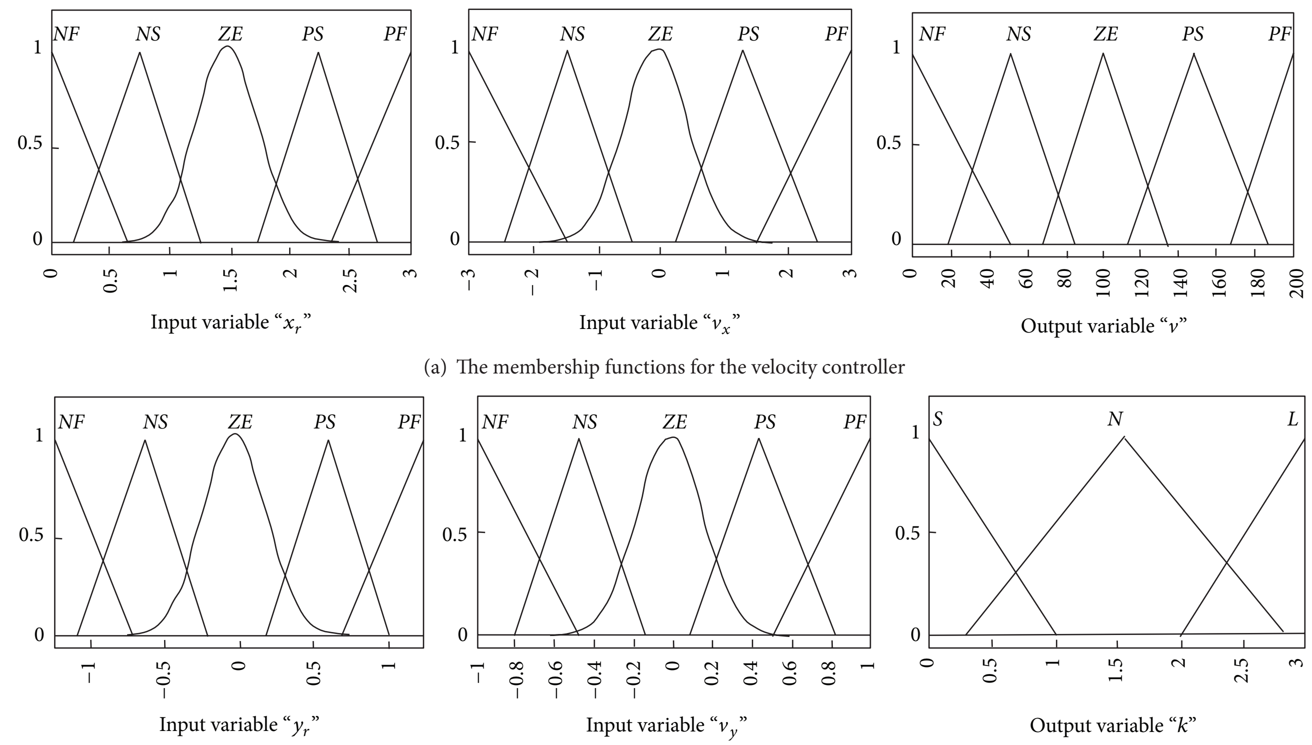
v		v_x				
		NF	*NS*	*Z*	*PS*	*PF*
x_r	*NF*	*VS*	*VS*	*S*	*S*	*Z*
	NS	*VS*	*S*	*S*	*Z*	*Z*
	Z	*VS*	*Z*	*Z*	*F*	*F*
	PS	*Z*	*F*	*VF*	*VF*	*VF*
	PF	*F*	*VF*	*VF*	*VF*	*VF*

5.1. Fuzzy Based Linear Velocity Controller. Our task is to keep the robot in a safe distance from the person while both the robot and person are moving. The distance between the robot and person varies due to their motions. In order to achieve a success track, the robot should change its linear velocity according to the distance obtained from the detector. Therefore, a fuzzy based linear velocity controller is designed to adaptively adjust the robot's velocity.

For the controller, the distance x_r and the person's vertical velocity v_x are chosen as inputs and the linear velocity v is chosen as output. For the inputs, two kinds of membership functions are used: the triangular membership function is for the large distance and velocity and the Gaussian membership function is for the small distance and velocity. For the output, we choose the triangular membership function. The domains of these parameters are $x_r \in [0, 3]$, $v_x \in [-1, 1]$, and $v \in [0, 200]$. The membership functions for these parameters are shown in Figure 7(a).

The fuzzy logic is established based on the human knowledge, which is shown in Table 1. According to the fuzzy logic, an adaptive linear velocity is obtained to drive the robot. In the case in which the distance and the speed of the person are the largest ($x_r = PF, v_x = PF$), the linear velocity will be accelerated to the maximal value ($v = PF$) for following the person as soon as possible. In contrast, if the distance and the person's velocity are very small ($x_r = NF, v_x = NF$), the robot will be slowed down to avoid hitting the person ($v = NF$). The fuzzy based controller makes the robot adapt its linear velocity according to the distance between the robot and person and the person's speed.

5.2. The Fuzzy Based Turning-Gain Controller. As following evolves, the person often wanders from the center of the robot's field. In such a case, the robot should change its turning radius in time to make sure that the person is in



(a) The membership functions for the velocity controller

(b) The membership functions for the turning-gain controller

Figure 7: The membership functions for linear velocity controller and turning-gain controller.

the center of the robot's field. To implement the task, a fuzzy based turning-gain controller is designed, where the robot's turning gain is adjusted according to the direction between the person and robot and the person's horizontal velocity.

The inputs for the fuzzy based turning-gain controller are the direction y_r and horizontal velocity of the person v_y, respectively. The output is the turning-gain k. The membership functions of the parameters for the fuzzy based turning-gain controller are the same as that of the fuzzy based linear velocity controller, shown in Figure 7(b). The domains of these parameters are $y_r \in [-1.25, 1.25]$, $v_y \in [-1, 1]$, and $k \in [0, 3]$.

The fuzzy logic is designed according to the human knowledge to determine the robot's turning-gain k. In the case in which the person moves to the left ($y_r = PF$) at positive fast speed ($v_y = PF$), the robot will turn at a very large turning gain ($k = L$) to make the human appear in the center of the robot's field again. When the person moves to the right ($y_r = NF$) at the positive speed ($v_y = PF$), the robot should turn at a normal turning gain ($k = N$) for implementing the following task. The fuzzy logic for the turning gain is shown in Table 2.

6. Experimental Results

Our person tracking algorithm is conducted on the Pioneer 3-DX robot.

Table 2: The fuzzy logic for turning gain controller.

k		v_y				
		NF	*NS*	*Z*	*PS*	*PF*
y_r	*NF*	*L*	*L*	*N*	*N*	*N*
	NS	*L*	*L*	*N*	*S*	*S*
	Z	*N*	*N*	*S*	*N*	*N*
	PS	*S*	*S*	*N*	*L*	*L*
	PF	*N*	*N*	*N*	*L*	*L*

6.1. User Is Moving but Robot Is Still.

6.1. User Is Moving but Robot Is Still. In this set of experiments, our method is compared with the color-texture based object representation algorithm [7]. These methods are evaluated on the color and depth image sequences captured from a still Kinect. The robot with the Kinect is still and a given person moves at about 1~3 m in front of the robot. In the following process, the given person moves here and there, and another person will pass by and occlude the given person. The comparison results are shown in Figure 8. The results in the first row are obtained by using the color-texture based algorithm; these in the second row are for our proposed method. When there is occlusion, the color-texture based method often fails to locate the person and loses him after occlusion. Using our method, the occlusion problem can be detected by analyzing the depth histograms and patches' similarity. Once the occlusion is detected, the method can recognize the given person by using the unoccluded patches' appearance model

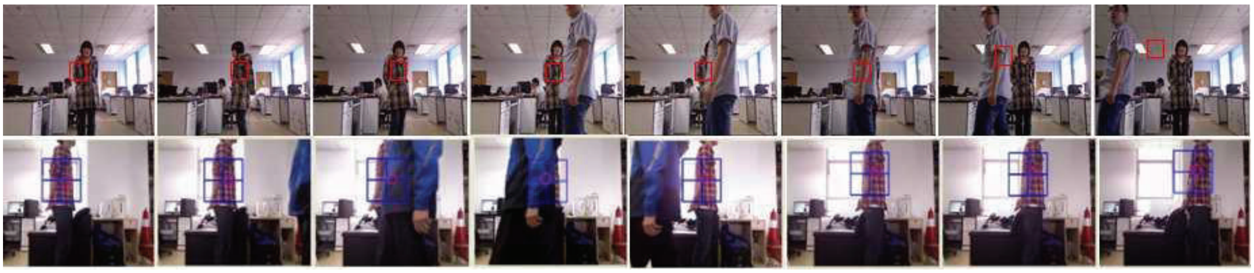


Figure 8: The tracking result using CT algorithm and our method when user is moving but robot is still.

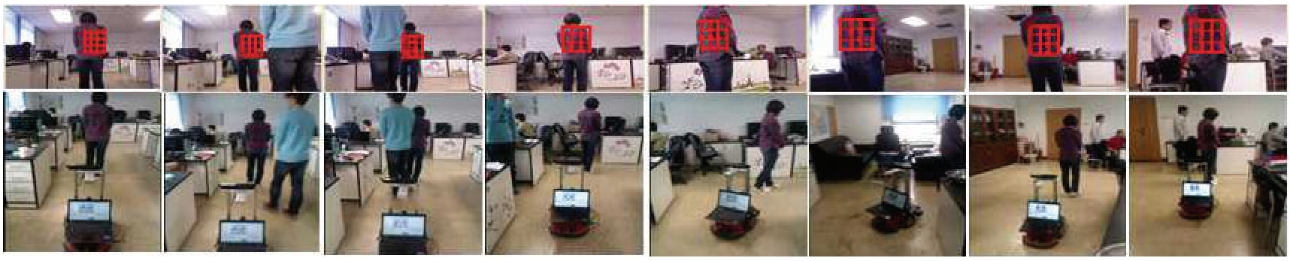


Figure 9: The tracking result using CT algorithm and our method when both of the user and the robot are moving.

represented by depth-color and depth-texture histograms. The results show that the patches based algorithm is of benefit to the occlusion problem. Furthermore, taking advantage of depth information, the person representation makes the foreground-background segmentation much easier.

6.2. Both the User and Robot Are Moving. In this section, our method is evaluated on a moving robot. As tracking evolves, there are occlusion, turning, appearance changes, and motion of both the robot and the given person. The tracking results are shown in Figure 9. The method tracks the person by adopting a patches based multicues detector and a MEKF tracker. In the case in which there is partial occlusion, the person is successfully detected by performing our method. When the person is fully occluded, the MEKF predicts the position of the person. Furthermore, an update strategy is adopted for updating the appearance representation in the tracking process. The experiment results illuminate that our method performs well in case of occlusion and appearance variations.

6.3. Robot Following Based on FZ-IGS. In this section, the performance of the presented FZ-IGS is evaluated. The given person's 3D position (x_r, x_y, x_z) was obtained from the data provided by the Kinect and was sent to the robot's controllers. The fuzzy based velocity controller changes the linear velocity according to the fuzzy logic. When there are variations in terms of distance between the robot and person (x_r) or the person's vertical speed (v_x), the linear velocity will accordingly change to make sure that the person is in a safe distance from the robot. Similarly, the turning-gain controller determines the turning gain according to the person's direction (x_y) and his speed (v_y) to keep the person in the center of the field of view of the robot. The paths for a person following robot are illustrated in Figure 10. For Figure 10(a), the red symbols "+" denote the path of the person, while the blue symbols "∘" are for the path of the robot. "0" is the start point of the person. In the beginning, the robot is still and the person is moving. When the distance between the person and the robot is larger than the safe distance (at about the point $x = 2000$, $y = 0$), the robot starts to follow the person. The results show that the robot can follow the person in a safe distance and keep him at the center of its FOV. In the case in which the distance or direction changes, the robot can vary its linear velocities and the turning gain to make sure that the robot can follow the robot stably. Figures 10(b) and 10(c) show the vertical distance and horizontal distance between the robot and the target according to time t, respectively. The results show that our method can guarantee that the robot tracks the person in a safe distance.

7. Conclusion

In this paper, we developed a new person tracking algorithm for a mobile robot. The paper exhibited four contributions. The first contribution concerned the person representation algorithm based on the fusion of multicues including depth-color histograms and depth-texture histograms. The color and texture information complement each other which improves the appearance's discrimination ability. The depth information easily discriminates the person from the background. The second contribution concerned patches based detection algorithm which divided the person into many patches. It could handle the partial occlusion problem by analyzing the unoccluded patches' similarity. The third contribution concerned the tracker MEKF which considers the motion of the robot and person. The fourth contribution

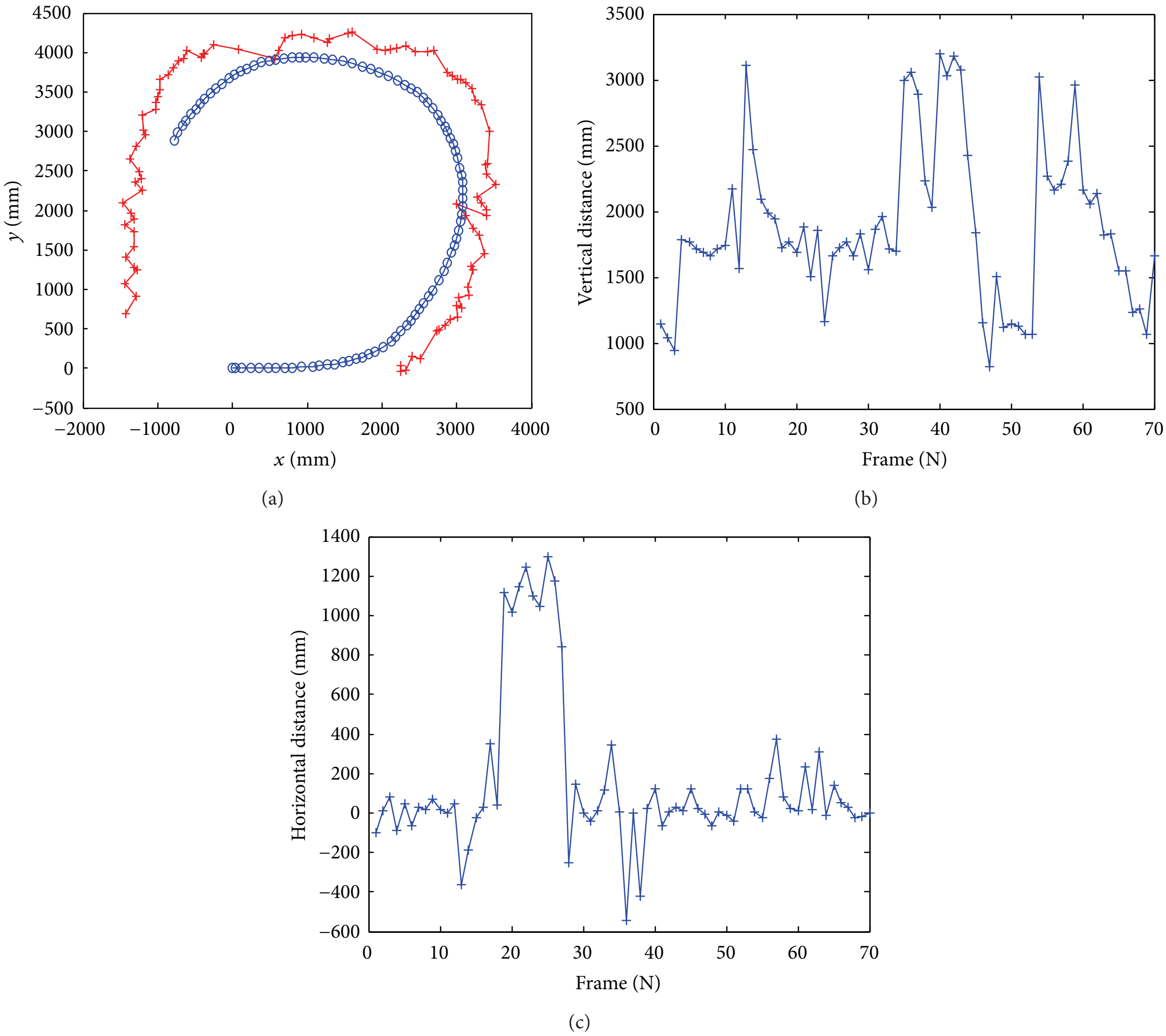


FIGURE 10: The robots path for following the target in the Lab.

concerned the fuzzy based intelligent controllers (FZ-IGS) which can adaptively change the linear velocity and turning-gain according to the person's positions obtained from the detector. The experimental results have demonstrated that the proposed method is able to track a person robustly and accurately. In the future, we will study the obstacle avoidance method in the tracking process.

Conflict of Interests

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

Acknowledgments

The research work is supported by National Natural Science Foundation (61175087), the Youth Foundation of Hebei Education Department, QN2014170, and Hebei province Science and Technology Support Program, 14275601D.

References

[1] M. Bansal, B. Matei, H. Sawhney, S.-H. Jung, and J. Eledath, "Pedestrian detection with depth-guided structure labeling," in *Proceedings of the IEEE 12th International Conference on Computer Vision Workshops (ICCV Workshops '09)*, pp. 31–38, Kyoto, Japan, October 2009.

[2] A. H. Mazinan and A. Amir-Latifi, "Applying mean shift, motion information and Kalman filtering approaches to object tracking," *ISA Transactions*, vol. 51, no. 3, pp. 485–497, 2012.

[3] Z. Zivkovic and B. Kröse, "An EM-like algorithm for color-histogram-based object tracking," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR '04)*, vol. 1, pp. I-798–I-803, IEEE, July 2004.

[4] J. F. Ning, L. Zhang, D. Zhang, and C. K. Wu, "Scale and orientation adaptive mean shift tracking," *IET Computer Vision*, vol. 6, no. 1, pp. 52–61, 2012.

[5] J. Ning, L. Zhang, D. Zhang, and C. Wu, "Robust object tracking using joint colour texture histogram," *International Journal of*

Pattern Recognition and Artificial Intelligence, vol. 23, no. 7, pp. 1245–1263, 2009.

[6] O. Zoidi, N. Nikolaidis, A. Tefas, and I. Pitas, "Stereo object tracking with fusion of texture, color and disparity information," *Signal Processing: Image Communication*, vol. 29, no. 5, pp. 573–589, 2014.

[7] S. Jia, S. Wang, L. Wang, and X. Li, "Robust human detecting and tracking using varying scale template matching," in *Proceedings of the IEEE International Conference on Information and Automation (ICIA'12)*, pp. 25–30, June 2012.

[8] L. A. Schwarz, A. Mkhitaryan, D. Mateus, and N. Navab, "Human skeleton tracking from depth data using geodesic distances and optical flow," *Image and Vision Computing*, vol. 30, no. 3, pp. 217–226, 2012.

[9] L. A. Schwarz, A. Mkhitaryan, D. Mateus, and N. Navab, "Estimating human 3D pose from time-of-flight images based on geodesic distances and optical flow," in *Proceedings of the IEEE International Conference on Automatic Face and Gesture Recognition and Workshops (FG '11)*, pp. 700–706, March 2011.

[10] L. Xia, C.-C. Chen, and J. Aggarwal, "Human detection using depth information by Kinect," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops (CVPRW '11)*, pp. 15–22, IEEE, Colorado Springs, Colo, USA, June 2011.

[11] A. Adam, E. Rivlin, and I. Shimshoni, "Robust fragments-based tracking using the integral histogram," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR '06)*, pp. 798–805, June 2006.

[12] S. M. S. Nejhum, J. Ho, and M.-H. Yang, "Visual tracking with histograms and articulating blocks," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR '08)*, pp. 1–8, 2008.

[13] M. Yang, J. Yuan, and Y. Wu, "Spatial selection for attentional visual tracking," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR '07)*, pp. 1–8, IEEE, Minneapolis, Minn, USA, June 2007.

[14] J. Kwon and K. M. Lee, "Highly nonrigid object tracking via patch-based dynamic appearance modeling," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 35, no. 10, pp. 2427–2441, 2013.

[15] N. Ouadah, V. Cadenat, F. Lerasle, M. Hamerlain, T. Germa, and F. Boudjema, "Multi-sensor-based control strategy for initiating and maintaining interaction between a robot and a human," *Advanced Robotics*, vol. 25, no. 9-10, pp. 1249–1270, 2011.

[16] M. Tarokh, P. Merloti, J. Duddy, and M. Lee, "Vision based robotic person following under lighting variations," in *Proceedings of the 3rd International Conference on Sensing Technology (ICST '08)*, pp. 147–152, December 2008.

[17] O. Azouaoui, N. Ouadah, I. Mansour, and A. Semani, "Fuzzy motion-based control for a bi-steerable mobile robot navigation," in *Proceedings of the 6th International Symposium on Mechatronics and Its Applications (ISMA '09)*, March 2009.

[18] L. Cao, C. Wang, and J. Li, "Robust depth-based object tracking from a moving binocular camera," *Signal Processing*, vol. 112, pp. 154–161, 2015.

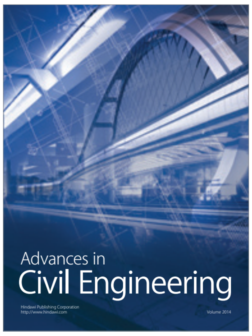
Journal of
Engineering

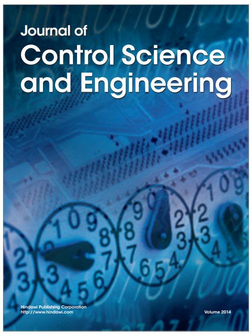
The Scientific World Journal

International Journal of
Rotating Machinery

Journal of
Sensors

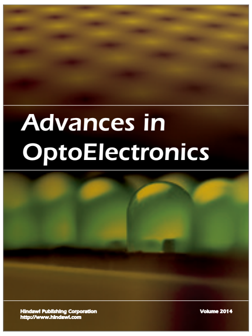
International Journal of
Distributed Sensor Networks


Advances in
Civil Engineering


Journal of
Control Science and Engineering

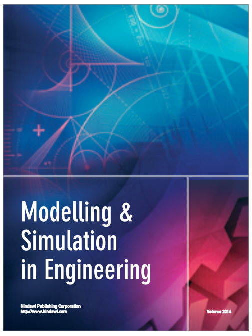
Journal of
Robotics

Journal of
Electrical and Computer Engineering

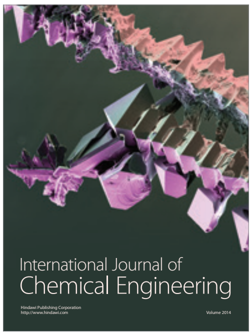

Advances in
OptoElectronics

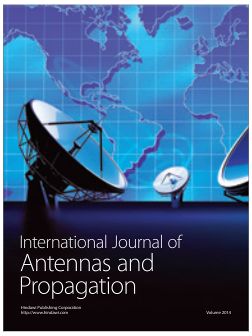
VLSI Design

International Journal of
Navigation and Observation

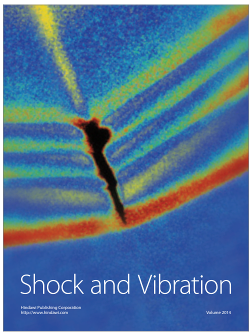

Modelling & Simulation in Engineering

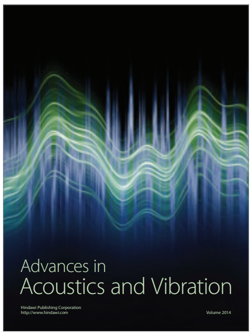
International Journal of
Aerospace Engineering


International Journal of
Chemical Engineering


International Journal of
Antennas and Propagation

Active and Passive Electronic Components


Shock and Vibration


Advances in
Acoustics and Vibration