京都大学
KYOTO UNIVERSITY

Title	People Detection Based on Spatial Mapping of Friendliness and Floor Boundary Points for a Mobile Navigation Robot
Author(s)	Tasaki, Tsuyoshi; Ozaki, Fumio; Matsuhira, Nobuto; Ogata, Tetsuya; Okuno, Hiroshi G.
Citation	Journal of Robotics (2011), 2011
Issue Date	2011
URL	http://hdl.handle.net/2433/187382
Right	© 2011 Tsuyoshi Tasaki et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.
Type	Journal Article
Textversion	publisher

Research Article

People Detection Based on Spatial Mapping of Friendliness and Floor Boundary Points for a Mobile Navigation Robot

Tsuyoshi Tasaki,[1] Fumio Ozaki,[2] Nobuto Matsuhira,[3] Tetsuya Ogata,[4] and Hiroshi G. Okuno[4]

[1] *Toshiba Corporate Research & Development Center, 1, Komukai-Toshiba-cho, Saiwai-ku, Kawasaki 212-8582, Japan*
[2] *Toshiba Corporation, Power Systems Company, 1-1, Shibaura 1-Chome, Minato-ku, Tokyo 105-8001, Japan*
[3] *Department of Engineering Science and Mechanics, Shibaura Institute of Technology, 3-7-5, Toyosu, Koto-ku, Tokyo 135-8548, Japan*
[4] *Graduate School of Informatics, Kyoto University, Yoshida-honmachi, Sakyo-ku, Kyoto 606-8501, Japan*

Correspondence should be addressed to Tsuyoshi Tasaki, tsuyoshi.tasaki@toshiba.co.jp

Received 14 July 2011; Revised 7 November 2011; Accepted 8 November 2011

Academic Editor: Danica Kragic

Navigation robots must single out partners requiring navigation and move in the cluttered environment where people walk around. Developing such robots requires two different people detections: detecting partners and detecting all moving people around the robots. For detecting partners, we design divided spaces based on the spatial relationships and sensing ranges. Mapping the friendliness of each divided space based on the stimulus from the multiple sensors to detect people calling robots positively, robots detect partners on the highest friendliness space. For detecting moving people, we regard objects' floor boundary points in an omnidirectional image as obstacles. We classify obstacles as moving people by comparing movement of each point with robot movement using odometry data, dynamically changing thresholds to detect. Our robot detected 95.0% of partners while it stands by and interacts with people and detected 85.0% of moving people while robot moves, which was four times higher than previous methods did.

1. Introduction

Mobile navigation robots are expected to move smoothly at big facilities such as big supermarkets, museums, and airports [1, 2]. Navigation robots are also expected to detect people that robots should navigate. Figure 1 shows a proposed navigation robot system. Our navigation robot system detects people who call the robot positively before navigation. When a person wants navigation service, our robot navigates the person to the destination by moving smoothly with detecting moving obstacles and avoiding them. Moving obstacles are dangerous, because the movements are not predicted easily. This paper focuses to describe people detection for our navigation robot system while robot stands by and moves.

People that robots want to detect while standing by are different from people that robots want to detect while moving. While standing by and interacting with people, the robot has to detect "people that call the robot positively" in order to offer a navigation service. While moving, it is important for a robot to detect all "moving people (obstacles)" around the robot in order to move smoothly.

Moreover, the feature of people detection is different, too. One different feature is related to calculation cycle. While standing by and interacting with people, slower people detection can be allowed comparing to the people detection for moving. Therefore, the robot can use multiple sensors that are used naturally in human-human interaction. For example, the sensors are cameras (eyes), microphones (ears), and tactile sensors (skins). While the robot moves, it needs fast people detection for safety and does not have to use all sensors that are used for interaction. Therefore, detection by one sensor is desirable while moving.

The other feature is related to localization accuracy (resolution). The robot does not need very high resolution
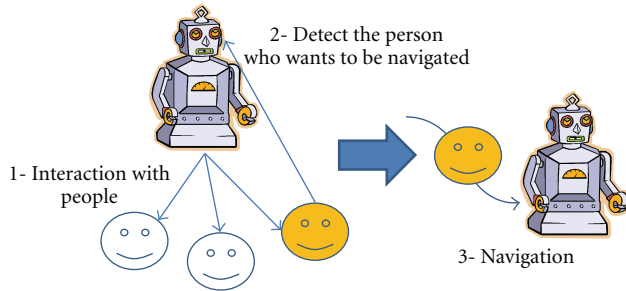
Figure 1: Our navigation robot system.

for interaction. While interaction, it is efficient to use appropriate resolution for interaction. On the other hand, while the robot moves, it needs high resolution and it wants to detect people accurately.

Recently, many works use distance measurement devices such as the Laser Range Finder (LRF) and stereo cameras [3–6] for people detection while interaction and moving. However, robots have to be equipped with more than one sensor when they classify all obstacles around them at once by the devices. Using many LRFs is expensive, and calibrating many cameras is troublesome. Using same kinds of multiple sensors is not desirable while moving. Moreover, these works do not deal with detecting people who call a robot.

In order to detect all moving people around a robot by using one sensor while the robot moves, an omnidirectional camera is useful. However, it is difficult to apply the previous methods that classify obstacles as moving people or not [7–10] to distorted omnidirectional images without modifying them to undistorted images. Even if we modify images, the previous methods do not work well because modified images lose a lot of information. Moreover, classifying obstacles as moving people or not by a mobile camera is more difficult than classifying them by a static camera.

We deal with two problems related to the people detection for a mobile navigation robot. One is detecting interaction partners who call a robot positively from among multiple people by using cameras, microphones, and tactile sensors. The other is classifying all obstacles around the robot as moving people or not by only one omnidirectional camera while the robot moves.

While robots stand by and interact with people, we have developed a method for detecting an interaction partner based on the degree of friendliness as mapped onto the "space", considering interaction distance and the range of multiple sensors for interaction.

For obstacle classification, we have also developed a new method that focuses on objects' floor boundary points where the robot can measure the distance from itself by only one omnidirectional camera. Our robot classifies a floor boundary point as a moving person when its movement is different from the robot's movement.

Solving these two problems, we have developed a mobile navigation robot which can select an appropriate person who calls the robot positively while robot stands by and can detect moving people while the robot moves. A contribution of

this paper is developing the people detection method for the navigation robot while the robot stands by and moves.

Section 2 describes our friendliness space map showing how friendliness is distributed in the space in order to detect an interaction partner. Section 3 describes the obstacle classification method based on tracking floor boundary points. In Section 4, we show the result of questionnaires and confirm an accuracy of our classification method. Section 5 concludes this paper.

2. Interaction Partner Detection by Friendliness Space Map While Interaction

2.1. Distance between the Robot and People While Interaction

2.1.1. Interaction Distance of People. When people interact with each other, the distance between them is associated with their degree of friendliness. Proxemics [11], which is a social psychology theory, says that two people interact at an appropriate physical distance from one another based on their relationship. In this theory, the interaction distance can be classified into roughly four groups: intimate, personal, social, and public.

   (i) Intimate distance (approximately 50 cm): people can communicate via physical interaction and express strong emotions.

   (ii) Personal distance (approximately 50–120 cm): people can talk intimately.

   (iii) Social distance (approximately 120–360 cm): people do not know each other well.

   (iv) Public distance (approximately 360 cm and more): people who have no personal relationship with each other can comfortably coexist at this distance.

These distances can be used to set the degree of friendliness between the robot and each person, which shows how positively each person calls the robot. The distances shown in parentheses are only typical ones. They depend on each person's personality and cultural background.

2.1.2. Effective Distance of Robot's Function. Since most functions and devices used by a robot are not effective for all distances, we assessed the effective distance for them. We investigated the effective distance of tactile recognition, speech recognition, sound source localization, and face localization, which are implemented into many robots as general functions.

(1) *Tactile Recognition:* Tactile recognition is done using tactile sensors, which are effective when people can touch the robot. The average length of a person's arm is up to 50 cm. This distance is similar to the intimate distance.

(2) *Speech Recognition:* To determine the range for speech recognition, we place a speaker in front of a robot at every 50 cm from 50 cm to 3.0 m and played 200 words of the ATR phonetically balanced corpus [12]. The results

TABLE 1: Relationship between distance and function.

Intimate distance	Personal distance	Social distance
Tactile recognition		
Speech recognition	Speech recognition	
Face localization	Face localization	Face localization
Sound localization	Sound localization	Sound localization

of isolated word recognition using Julian [13], general Japanese automatic speech recognition software, show that the recognition rate is more than 85% for distances less than 1.5 m. Automatic speech recognition was found to be effective up to around 1.5 m.

(3) *Sound Source Localization:* A well-known sound source localization function uses the Interaural Phase Difference (IPD) and Interaural Intensity Difference (IID) [14]. The effective distance of sound source localization on average and the standard deviations were estimated in our laboratory. Three directions were evaluated separately. The horizontal direction was specified from right (0 deg) to left (180 deg), and the center was 90 deg. The localization errors were small (less than 3 deg) for distances less than about 3 m. Therefore, sound source localization should be stable up to around 3 m.

(4) *Face Localization:* We use MPIsearch [15] for face localization. The robot can measure the distance and direction. MPIsearch requires an image at least 12 by 12 pixels to detect a face. Such images correspond to a distance of 4.0 to 5.0 meters. In general, the effective distance of face localization is up to the public distance. This distance is decided by the size of template, the size of captured image, and the angle of view. The distance is not related to a selected algorithm very much.

Detail discussions of effective distances are described in [16].

2.1.3. Interaction Distance and Effective Distance of Functions. The relationship between the interaction distance and the effective distance for the four functions is shown in Table 1. As shown in the table, effective distance for the functions can correspond to the interaction distance effectively.

2.2. Friendliness Space Map

2.2.1. Design Friendliness Space Map. The sensor functions a robot can use effectively differ depending on the distance between the robot and each person. In other relational studies, the robot always used all sensors and interacted with people by focusing on the people. In our study, the robot interacted with people by focusing on the "space" of the people. In particular, the robot acted based on the space around the robot, segmented as described in Table 1.

Given the size of a person's face and the accuracy of the robot's functions, the direction element of space must be segmented to some extent. We segmented the space every 15 degrees based on the average size of the human face (16 cm ×
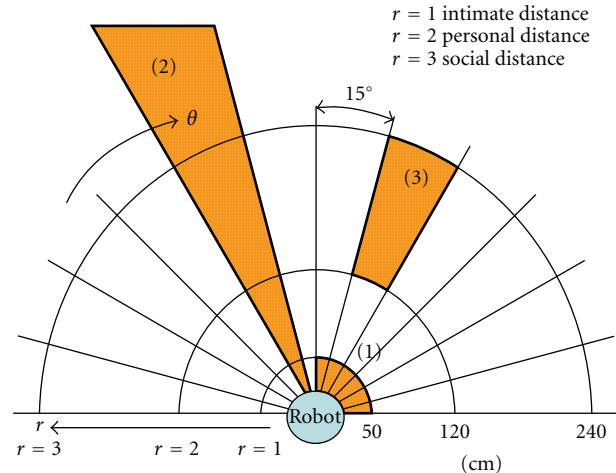

FIGURE 2: Friendliness space map and effective area of functions.

23 cm) and the errors of functions within the personal distance.

To identify the intimate space for the robot to interact with, we defined polar coordinates as shown in Figure 2. These coordinates, which are segmented into cells, are called a "Friendliness Space Map." Our robot calculates the "friendliness" of cell (r, θ) using information about the location of people and comfortable/uncomfortable stimuli. To calculate the friendliness which shows how positively people call the robot, when a function is initiated by sensor input, our robot calculates the Human Existence Degree (HED), which shows whether people exist or not, of cells within the effective area of each function. For example, three areas where our robot calculated the HED are shown in Figure 2: (1) in the case the right side of the robot is touched, (2) in the case the robot detects sound, and (3) in the case the robot detects face.

The effects of detecting the interaction partner using this map are as follows.

(i) Since a robot can change its motion and select an interaction partner based on the friendliness of various spaces, it can attract people while it stands by.

(ii) The action selection based on space can also be applied to various other objects.

2.2.2. Definition of Human Existence Degree by Integration of Functions. In each cell on the map, the HED is calculated by taking advantage of the integrated functions. When a function k locates a person at time t_{k0}, it calculates the HED, $L_{k,t,r,\theta}$, of cell (r, θ) within the effective function area at time t, as shown in (1). The k ($k = 1, 2, 3$) is the functions, and d_k is the damping ratio which is decided based on the degree of confidence obtained by previous experiments of each function. The damping ratio introduced for expressing the accuracy of sensing becomes low as time goes by. Here, L is related to only time t for simplicity, though L may be related to many parameters. t_{k0} is renewed every time function k operates:

$$L_{k,t,r,\theta} = \exp[-d_k(t - t_{k0})]. \tag{1}$$

The HED calculated by integration of all functions, $E_{t,r,\theta}$, of cell (r, θ) at time t is defined as the sum of the HED of each function:

$$E_{t,r,\theta} = \sum_{k=1}^{3} L_{k,t,r,\theta}. \qquad (2)$$

2.2.3. Shift in Friendliness by Stimulus. The cells on the Friendliness Space Map are affected by the kind of stimulus which shows positivity or negativity. Our robot recognizes two kinds of stimuli by using tactile recognition. One is uncomfortable stimuli which show negativity, such as hitting the robot's head or touching the robot's bust. The other is comfortable stimuli which show positivity, such as patting the robot's head. These stimuli are decided by human-human interaction when a person selects interaction partner. Comfortable stimuli are used to call person. On the other hand, uncomfortable stimuli are used to just tease.

Since tactile recognition cannot localize people precisely, we assume that the person delivering the stimulus is in the cell with the highest HED within the intimate distance. That is, it is cell $(1, \theta)$, as obtained using

$$\theta = \arg\max_{\theta} E_{t,1,\theta}. \qquad (3)$$

If the stimulus occurs at time t_{C0}, we define the Comfortable Degree (CD), $C_{t,1,\theta}$, of cell $(1, \theta)$ selected at time t as shown in (4). Here, d_C denotes the damping ratio, and v denotes the kind of stimulus. When the stimulus is comfortable, v is 1. When the stimulus is uncomfortable, v is -1. t_{C0} is renewed every time a stimulus is received:

$$C_{t,r,\theta} = v \times \exp[-d_C(t - t_{C0})]. \qquad (4)$$

2.2.4. Definition of Friendliness. The Friendliness Space Map is renewed and consists of both the HED and the CD obtained using the robot's functions. The friendliness, $I_{t,r,\theta}$, of cell (r, θ) at time t is defined as the sum of the HED and the CD as shown in (5), where W_L and W_C correspond to the weights of the HED and the CD, respectively. In this time, we make W_C bigger than W_L because we want a robot to be sensitive to the stimulus:

$$I_{t,r,\theta} = W_L \times E_{t,r,\theta} + W_C \times C_{t,r,\theta}. \qquad (5)$$

3. Moving People Detection by Classifying Obstacles Based on Floor Boundary Points While Moving

3.1. Floor Boundary Points Detection

3.1.1. Floor Detection by Ward's Clustering. We use floor colors for floor detection because floor colors are generally simple. Previous works use the Gaussian Mixture Model (GMM) for specific color detection [17]. The GMM can detect many specific colors, increasing a number of a mixed Gaussian. However, we have to evaluate the GMM many times in order to decide parameters such as the number of

mixed Gaussian. Therefore, it is difficult for robots to apply the GMM to various environments quickly and accurately just after they start up.

Our robot learns representative colors of the floor by itself based on the distribution of floor color data without prior setting. Considering the distribution, our floor detection method can adjust more easily than the GMM can and detects the floor as accurately as the GMM does. Here, in order to detect the representative colors of the floor, we assume that our robot is activated in the free space. Moreover, we use Ward's clustering [18], which is one of the hierarchical clustering methods. Our robot selects the representative colors by Ward's clustering as follows.

(1) Our robot takes an image and gets N color data from pixels to which the close area around it is projected. In an initial state, each datum shows a representative color. A cluster of color data that are similar to the representative color i is denoted by C_i.

(2) We choose two clusters C_1 and C_2 that minimize D as shown in (6) and create a new cluster C_k that consists of the data in both C_1 and C_2. Let c_i denote an average color vector in the cluster C_i:

$$D(C_1, C_2) = d(C_1 \cup C_2) - d(C_1) - d(C_2),$$
$$d(C_1) = \sum_{x \in C_i} \|x - c_i\|. \qquad (6)$$

(3) In step 2, when C_k satisfies both (7) and (8), it is decided that c_k is the representative color and data in C_k are not used for following loops. When C_k satisfies only (7), data in C_k are just not used for following loops. T_D and T_N are constant thresholds, $|C_k|$ is a number of the data in C_k:

$$\min_{k \neq i} D(C_k, C_i) > T_D, \qquad (7)$$

$$|C_k| > T_N. \qquad (8)$$

(4) Steps 2 and 3 continue until all data are not used.

Because Ward's clustering considers the distribution of data, each cluster is identified easily by Mahalanobis distance. A color datum I is classified as floor color when we find a C_o that satisfies (9). μ_o, Σ_o, and σ denote an average vector, a covariance matrix of data in C_o, and a threshold, respectively:

$$\sqrt{(I - \mu_o)^T \Sigma_o^{-1} (I - \mu_o)} < \sigma. \qquad (9)$$

When a robot uses an omnidirectoinal camera mounted on its head, the floor is projected to around the image center. Therefore, our robot classifies the pixels from center to outer by applying (9). If our robot finds continuous p pixels that do not satisfy (9), a floor boundary point is detected at the position where the first pixel in p pixels is located. These points show the boundary between the free space and obstacles and can be tracked easily. We have already confirmed that our floor detection method can work well on the supermarket floor [19]. However, not all points locate on the boundary between the floor and obstacles. We change dynamically by the method described in Section 3.2.2.

3.1.2. Transforming Coordinates of Floor Boundary Points from Image Coordinates to Robot Coordinates.

In the case of using an omnidirectional camera incorporating a hyperbolic mirror, a position (X, Y, Z) on the robot coordinates is projected to a position (x, y) on the image coordinates as follows [20]. Constants b and c denote proper parameters of the mirror, and f denotes a focal distance:

$$x = \frac{Xf(b^2 - c^2)}{(b^2 + c^2)(Z - c) - 2bc\sqrt{X^2 + Y^2 + (Z - c)^2}},$$

$$y = \frac{Yf(b^2 - c^2)}{(b^2 + c^2)(Z - c) - 2bc\sqrt{X^2 + Y^2 + (Z - c)^2}}. \quad (10)$$

Many robots are equipped with an omnidirectional camera, and they can measure or know the distance from the floor to the camera while they are moving [21, 22]. Therefore, with regard to floor boundary points, the variable Z in (10) becomes constant, and we can measure the distance from the robot to floor boundary points by applying (10).

In order to decide the parameters Z, b, c, and f, we have drawn cross-stripes on the floor as shown in Figure 3(a). n pairs of (X_a, Y_a) and (x_a, y_a) are acquired from the image to which n cross-points are projected. Here, (X_a, Y_a) and (x_a, y_a) denote the position of the cross-point a on the robot coordinates and the image coordinates, respectively. Using n pairs, parameters that minimize the evaluation function F_v as shown in (11) are decided by the downhill simplex method:

$$F_v$$
$$= \sum_{a=0}^{n-1} \left| x_a - X_a f \frac{b^2 - c^2}{(b^2 + c^2)(Z - c) - 2bc\sqrt{X_a^2 + Y_a^2 + (Z - c)^2}} \right|$$
$$+ \sum_{a=0}^{n-1} \left| y_a - Y_a f \frac{b^2 - c^2}{(b^2 + c^2)(Z - c) - 2bc\sqrt{X_a^2 + Y_a^2 + (Z - c)^2}} \right|. \quad (11)$$

For confirmation of parameters, a bird's-eye image is created by using the decided parameters. Figure 3(b) shows the bird's-eye image. The lines that make cross-stripes on the floor are not distorted, because the decided parameters are corrected. Here, 1 pixel in this bird's-eye image denotes about 5.0 cm in the real world.

3.2. Obstacle Classification by Floor Boundary Points

3.2.1. Classification Equation.

A floor boundary point m on the image at time $t - dt$ is detected by the method as shown in Section 3.1. dt depends on a processing speed. If the point m can be tracked from $t - dt$ to t correctly, the position of m at t is located correctly on the image at t. Here, we use Lucas Kanade tracker algorithm with image pyramid representation [23] as a tracking method. The tracking method works well even in the omnidirectional image as shown in [24].

It is easy to transform the coordinates of m at $t - dt$ and t from the image coordinates to the robot coordinates
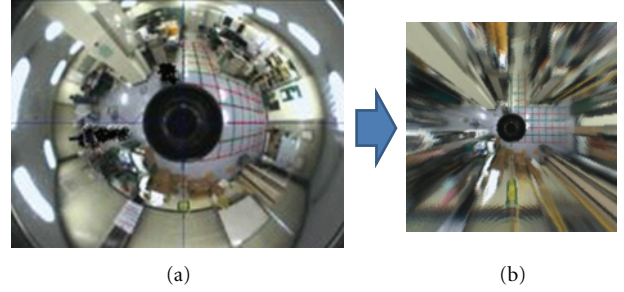

(a)   (b)

FIGURE 3: The omnidirectional image of the cross-stripes on the floor (a) and the bird's eye images (b).

$(X_m, Y_m)^{(t-dt)}$ and $(X_m, Y_m)^{(t)}$ by referring to the bird's-eye image. The relative position $(dX, dY, d\Theta)$ from $t - dt$ to t is estimated by odometry data. $d\Theta$ is based on the direction from the center of the robot to the front of the robot at $t - dt$. When m is located at the boundary between a static obstacle and the floor, $(X_m, Y_m)^{(t)}$ is calculated by $(dX, dY, d\Theta)$ and $(X_m, Y_m)^{(t-dt)}$, as shown in (12):

$$\begin{pmatrix} X_m \\ Y_m \end{pmatrix}^{(t)} = \begin{pmatrix} \cos d\Theta & -\sin d\Theta \\ \sin d\Theta & \cos d\Theta \end{pmatrix} \begin{pmatrix} X_m \\ Y_m \end{pmatrix}^{(t-dt)} + \begin{pmatrix} dX \\ dY \end{pmatrix}. \quad (12)$$

When m is located at the boundary between a moving obstacle (person) and the floor, (12) is not satisfied. Therefore, we can regard (12) as a Classification Equation (CE); that is, the floor boundary point m can be classified as a static obstacle or a moving one by confirming whether (12) is satisfied or not. Actually, (12) includes a small error ε depending on an image resolution and uncertainty in sensing, which is ignored.

The following conditions should be satisfied in order to regard (12) as the classification equation.

(1) Floor boundary points have to be located at the boundary between obstacles and the floor correctly in the image.

(2) Floor boundary points have to be tracked correctly.

(3) Camera parameters have to be decided correctly.

(4) Odometry has to be calculated correctly.

Condition 4 is satisfied in the general environment, because the odometry is comparatively correct during short movement. Figure 3 verifies that parameters are not so bad that condition 3 is satisfied, too. Floor boundary points can be tracked easily and tracking is not a major problem by the tracking method [25] when they are detected accurately, because they are located at the boundary where the colors change significantly. However, floor boundary points cannot always be detected correctly by using only the floor colors in various environments. We apply the result of confirming whether the CE is satisfied or not to the floor detection method.

3.2.2. Obstacle Classification.

The CE is satisfied as long as floor boundary point m is located on the floor. One of the

reasons why m is not located on the floor is that the threshold σ in (9) is inappropriate. When the position of m does not satisfy the CE, σ is too large or m shows a moving obstacle. For confirmation, new floor boundary point m' is detected by decreasing σ in the direction where m is located to $\sigma - d\sigma$. The parameter $d\sigma$ should be small so that the robot does not narrow the floor area. The new floor boundary point m' is tracked from t to $t - dt$ and classified by confirming the CE again. When the position of m' satisfies the CE, our robot regards m as a static obstacle. Moreover, the position of m is changed to the position of m'. Conversely, if it is not satisfied, m is regarded as a moving obstacle. Our method changes the parameter dynamically by the result of the CE. For example, in Figure 4, the position of floor boundary point A located at the boundary between the floor and a static obstacle satisfies the CE. The point B that is not located at the boundary does not satisfy the CE. Therefore, B creates a new floor boundary point B' and B' is tracked from t to $t - dt$. Using the result of tracking, our robot confirms whether B' satisfies the CE or not. Because B' is located at the boundary, B' satisfies the CE in this case. Therefore, the position of B is changed to the position of B' and B is classified as a static obstacle. The point C located at the boundary between a moving obstacle and the floor also does not satisfy the CE. The point C creates a new point C' and its position is confirmed. Because $d\sigma$ is small, the point at the boundary does not create a new point far from the original point. The position of C' does not satisfy the CE in this case, and C is regarded as a moving obstacle.

If the threshold is low at the beginning of the robot's activation, all points are located on the floor. However, they are located between the boundary and the robot, and free space looks very small. Our classification method first uses high thresholds and detects the boundary that is a little larger than the true boundary. Moving and confirming the CE refine the threshold of each direction where the floor boundary point classified as a moving obstacle is located. Finally, the robot adapts the threshold of each direction and makes it possible to locate and classify obstacles accurately. When the illumination and floor color change, our robot adapts the threshold again.

4. Evaluation

4.1. Our Robot and Experimental System. Our people detection method is implemented on our robot called ApriTau as shown in Figure 5 left. It has a vehicle that can acquire the odometry data. An omnidirectional camera is mounted on the top of its head and does not move with the head motion. Taking images while moving, it synchronizes the odometry data. ApriTau takes images whose size is 320×240 pixels continuously at 30 fps. It has microphones and touch sensors. It can detect people by using these sensors based on the method as shown in Section 2.2.1. It can move its head and gaze at the interaction partner.

Figure 6 shows our obstacle classification system while robot moves. The inputs are continuous omnidirectional images. The outputs are the results of the classification of each direction. The system detects 360 floor boundary points
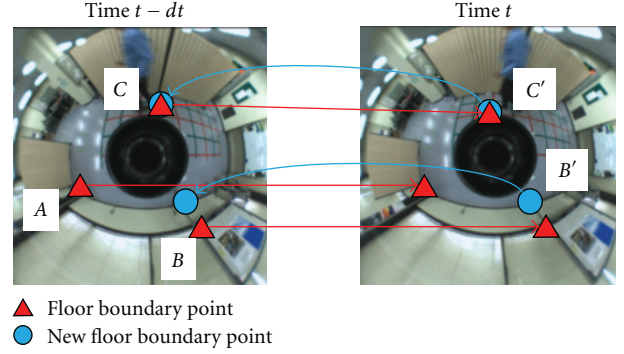

FIGURE 4: The example of classification process by using floor boundary points. Triangles are tracked from the left image to the right image. Circles are tracked from the right image to the left image. The robot moves by 20 cm.
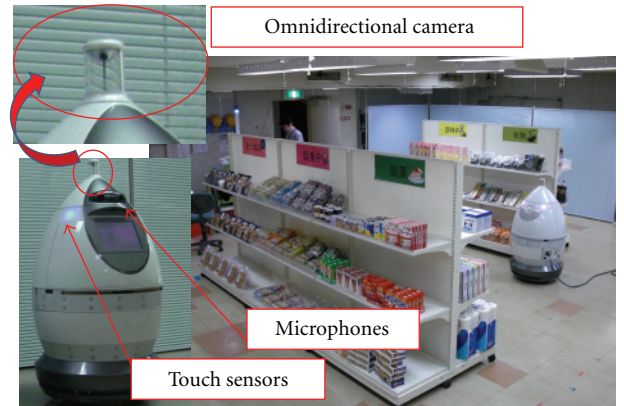

FIGURE 5: ApriTau and experimental setting.

using the result of tracking previous points or the floor detection method in image at $t - dt$. Red squares or blue points are floor boundary points in Figure 7. 360 points are detected every one degree. These points are tracked and classified. In Figure 7, blue points and red squares are classified as static obstacles and moving obstacles (people), respectively. Most of them are located at the boundary between the floor and obstacles. A red line is drawn from the image center to the average of red points' positions. This system integrates floor boundary points which are classified as moving obstacle like the red line, when points which are classified as moving obstacles are located near (less than 10 degree) the other points which are classified as moving obstacles. In order to learn the floor color, our robot is activated in the free space whose size is 1.0 (m) by 2.0 (m). We assume that we can find the space before opening facilities.

In these experiments, the thresholds T_D and T_N and the parameter p as shown in Section 3.1 for floor detection are 18000, 10 pixel, and 3, respectively. These thresholds and parameters are decided experimentally, considering the resolution of the image.
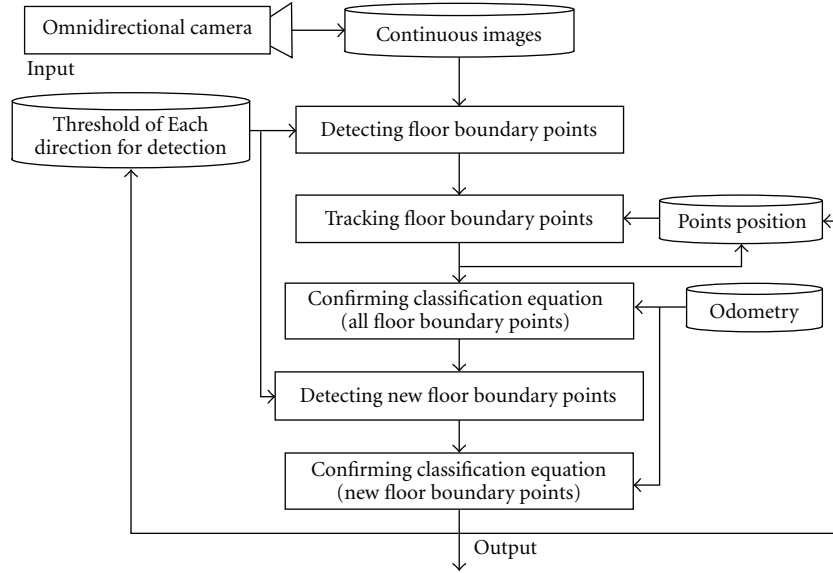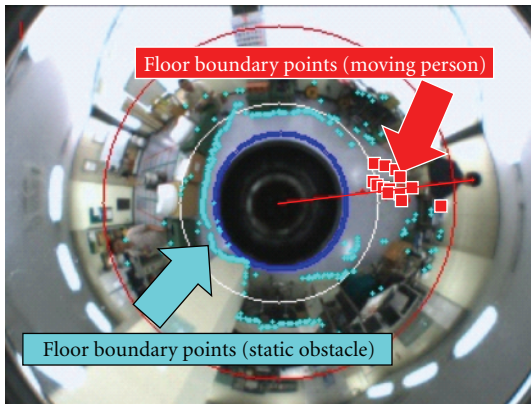
Figure 6: The whole system of classification.

Figure 7: The output of classification system.

4.2. Confirmation of Detecting People While the Robot Stands by and Interacts with People

4.2.1. Aim and Sequence of Experiment. We investigated whether our method detects interaction partner while the robot stands by and interacts with people. We asked 4 people to interact with our robot freely. Our robot looks at the highest friendliness place and talks with people by using only simple words. Two labelers observe their interaction and select interaction partners whom our robot should interact with on the second time base.

We evaluate our method by two values $E1$ and $E2$ as shown in (13) and (14):

$$E1 = \frac{T_{\text{rob}}}{T_{\text{lab}}}, \quad (13)$$

$$E2 = \frac{T_{\text{exist}}}{T_{\text{out}}}. \quad (14)$$

T_{lab} shows the duration when two labelers select same partners. T_{rob} shows the duration when two labelers and our robot select same partners. T_{out} shows the duration when our robot outputs detecting people. T_{exist} shows the duration when our robot outputs detecting people correctly.

4.2.2. Result and Discussion. The experimental results show that $E1$ denotes 0.95 and $E2$ denotes 0.87. We think that $E1$ is high enough to detect people who call robots. $E1$ is higher than $E2$, which shows that our robot can especially select people whom humans (labelers) can select by only observing the interaction.

One of the reasons why $E2$ is a little low is that people do not always call the robot. Therefore, both our robot and labelers do not select the person to interact with. We think that it is not a problem because our system aims to detect people who call robots.

4.3. Evaluation of Obstacle Classification

4.3.1. Aim and Sequence of Experiment. In order to confirm the effectiveness of changing the threshold σ dynamically based on the result of the CE, we compared the classification ratio of our method with that of a simple method using a constant threshold and that of a previous method. As the previous method, we use the method that modifies omnidirectional images to general images and detects movements that is different from movements of background, as shown in [26]. The color of the floor is not complex. The experimental steps are as follows.

(1) ApriTau and another robot move on the given route. They pass each other.

(2) ApriTau takes images synchronized with odometry data continuously while moving.

Table 2: The classification ratios.

Method	Recall ratio	Precision ratio	F value
Previous	0.18 (3/17)	0.25 (3/12)	0.21
Simple	0.63 (10/16)	0.13 (10/79)	0.21
Ours	0.94 (17/18)	0.77 (17/22)	0.85

(3) The images and the odometry data are input to the systems of our method, the simple method, and the previous method. Note that although same data are input to three systems, each system processes some of them because of the difference of the processing speed.

(4) The classification ratios of our method, the simple method, and the previous method are calculated by outputs.

In this experiment, the classification ratio is the F value calculated by the recall ratio R and the precision ratio P as shown in (15). Here, N_A, N_O, and N_C show the number of images to which another moving robot is projected, the number of obstacles the system classified as moving obstacles, and the number of moving obstacles the system outputs and locates correctly, respectively:

$$R = \frac{N_C}{N_A}, \qquad P = \frac{N_C}{N_O}, \qquad F = \frac{2RP}{(R+P)}. \qquad (15)$$

4.3.2. Result and Discussion. The classification ratios of three methods are shown in Table 2. The classification ratio of our method is 4 times higher than that of the simple method and that of the previous method. In particular, the improvement of the precision ratio affects the F value. One of the reasons why the precision ratio of our method is much higher than that of the simple method is that ApriTau can select floor boundary points showing candidates of moving obstacles by the CE and relocate points correctly by strengthening the threshold detecting each point. The result shows that the accuracy of locating points greatly affects classification ratio. One of the reasons why the F value of the previous method is low is losing information by changing omnidirectional images to general images. Another reason is that ApriTau and another robot pass each other. The movement of another robot is similar to the movement of background, and the previous method cannot detect another robot.

However, the precision ratio of our method is a little low for robots' smooth movement. In this paper, we assume that errors of tracking points are very small, which is certainly correct to some extent for the image coordinates. In the case of omnidirectional camera image, the distance resolution changes depending on the distance from the image center. It is very low for a distant place. Tracking errors of a few pixels become errors of a few meters for the world coordinates. Because of errors of a few meters, (12) does not work as the CE. When the floor boundary point is located at a position distant from the center of the image, we have to track it for a longer time and use its average movement. Moreover, it also might be effective to use adaptive scheme instead of the fixed parameter $d\sigma$ as shown in Section 3.2.2.
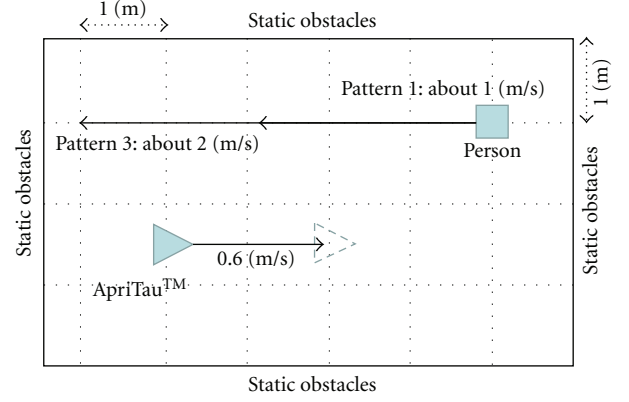

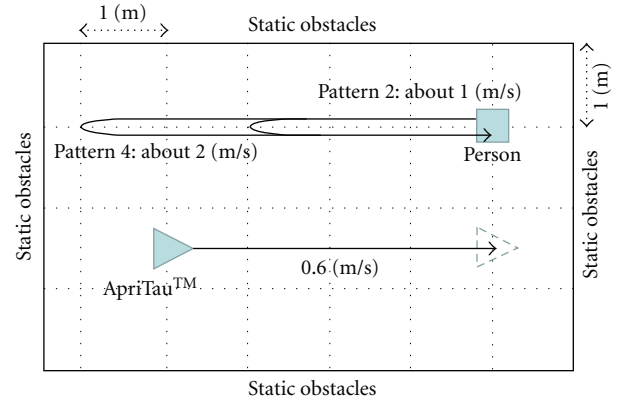Figure 8: The experimental setting (Pattern 1 and 3).

Figure 9: The experimental setting (Pattern 2 and 4).

4.4. Evaluation of Moving People Detection

4.4.1. Aim and Sequence of Experiment. In order to confirm our method detects moving people, we calculate the classification ratio in various patterns. In this experiment, a person and ApriTau move on the given route as shown in Figures 8, 9, and 10. In order to confirm basic ability of our method, ApriTau and one person go straight and rotate. As same as the experiment in Section 4.3, ApriTau takes images synchronized with odometry data and the classification ratio of our method is calculated.

4.4.2. Result and Discussion. The classification ratios in each pattern are shown in Table 3. The classification ratios in the case of the person walking (Patterns 1 and 2) are higher than 0.77, which is as high as the classification ratios in Section 4.3. The classification ratios in the case of the person running (Patterns 3 and 4) are a little low. One of the reasons why the classification ratios are a little low is that the boundary between the running person and the floor is more complex than the boundary between the walking person and the floor. The complex boundary can make robots fail to detect floor boundary points accurately. We think that increasing floor boundary points can solve this problem.
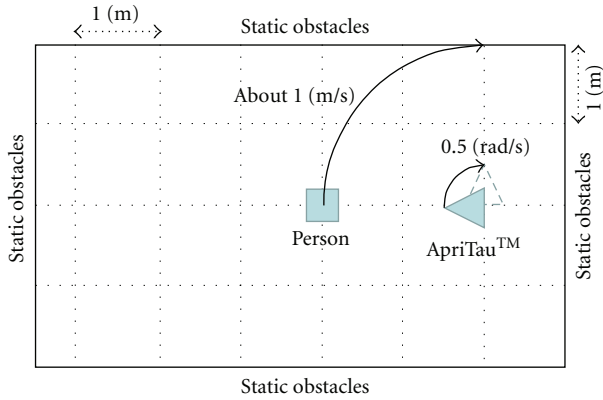
Figure 10: The experimental setting (Pattern 5).

Table 3: The classification ratios in various patterns.

Pattern	Recall ratio	Precision ratio	*F* value
1	1.00 (21/21)	0.64 (21/33)	0.79
2	0.98 (42/43)	0.64 (42/67)	0.77
3	0.80 (16/20)	0.64 (17/29)	0.71
4	0.93 (42/45)	0.59 (42/74)	0.72
5	0.93 (13/14)	0.57 (13/23)	0.71

The classification ratio in the case of the robot rotation (Pattern 5) is also a little low. One of the reasons why the classification ratio is a little low is that tracking area in the image in the case of rotation changes more than tracking area in the case of straight transition (Patterns 1–4) does. Changing tracking area very much makes robots fail to track the floor boundary points. Moreover, we need to synchronize the timestamps between odometry and images. We also think that it is effective to take into account uncertainty in sensing. The accuracy of odometry or tracking differs according to the robot movement. We have to use probabilistic method in the future work.

5. Conclusion

This work has dealt with two problems related to people detection that is needed for the navigation robot system. One is how robot detects the person who calls it positively while standing by in order to select a person. The other is how one moving omnidirectional camera detects all moving people around the robot while moving in order to move safely. Changing the people detection methods according to tasks of the robot, we aim to select the person who needs navigation and detect moving people while robot moves in particular for safety.

In order to solve the first problem, we have developed a people detection method based on the "friendliness space map," which focuses on the "space" rather than the person to find and select people who call our robot positively.

In order to solve the second problem, we have developed the new method that focuses on floor boundary points where one omnidirectional camera can measure the distance

from the robot. The points are detected by the floor detection method using Ward's clustering to find representative colors and Mahalanobis distance to identify floor colors. For detecting moving people, our robot tracks the floor boundary points. Comparing the robot's movement with floor boundary points' movement, our robot detects moving people and dynamically changes the threshold that the floor detection uses.

We performed three experiments. The first experimental result showed that our robot detects 95% of the person who calls the robot positively by using friendliness space map. In the second experiment, we confirmed the classification ratio increased to 85%, which was four times higher than that of a previous method. The third experimental result showed that our method could detect a moving person in various situations. In future work, we plan to evaluate our navigation system in a crowded place such as a real supermarket. (This paper is an extended version of a conference paper [27] with additional description of moving people detection and a navigation robot system.)
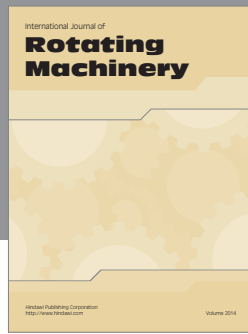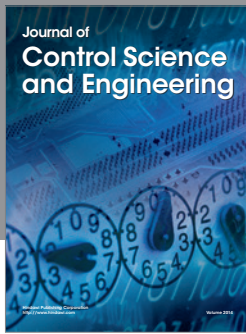
Acknowledgment

References

[1] S. Thrun, M. Bennewitz, W. Burgard et al., "MINERVA: a second-generation museum tour-guide robot," in *Proceedings of the IEEE International Conference on Robotics and Automation*, pp. 1999–2005, May 1999.

[2] R. Philippsen and R. Siegwart, "Smooth and efficient obstacle avoidance for a tour guide robot," in *Proceedings of the IEEE International Conference on Robotics and Automation*, pp. 446–451, September 2003.

[3] Z. Jia, A. Balasuriya, and S. Challa, "Sensor fusion based 3D target visual tracking for autonomous vehicles with IMM," in *Proceedings of the IEEE International Conference on Robotics and Automation*, pp. 1829–1834, April 2005.

[4] M. Weser, D. Westhoff, M. Hüser, and J. Zhang, "Multimodal people tracking and trajectory prediction based on learned generalized motion patterns," in *Proceedings of the IEEE International Conference on Multisensor Fusion and Integration for Intelligent Systems,*, pp. 541–546, September 2006.

[5] Z. Chen and S. T. Birchfield, "Person following with a mobile robot using binocular feature-based tracking," in *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 815–820, November 2007.

[6] S. Nishio, N. Hagita, T. Miyashita et al., "Sensor network for structuring people and environmental information," in *Cutting Edge Robotics*, pp. 367–378, InTech, 2010.

[7] G. Silveira, E. Malis, and P. Rives, "Real-time robust detection of planar regions in a pair of images," in *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 49–54, October 2006.

[8] Y. Pang, Q. Huang, W. Zhang, Z. Hu, A. H. Rajpar, and K. Li, "Real-time object tracking of a robot head based on multiple visual cues integration," in *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 686–691, October 2006.

[9] B. Jung and G. S. Sukhatme, in *Proceedings of the Conference on Intelligent Autonomous Systems*, pp. 980–987, 2004.

[10] G. Chivilò, F. Mezzaro, A. Sgorbissa, and R. Zaccaria, "Follow-the-leader behaviour through optical flow minimization," in *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 3182–3187, October 2004.

[11] E. T. Hall, *The Hidden Dimension*, Doubleday Publishing, 1990.

[12] H. Kuwabara, Y. Sagisaka, K. Takeda, and M. Abe, "Construction of ATR Japanese speech database as a research tool," Tech. Rep. TR-I-0086, 1989.

[13] A. Lee and T. Kawahara, "Recent development of open-source speech recognition engine Julius," in *Proceedings of the Asia-Pacific Signal and Information Processing Association Annual Summit and Conference*, pp. 131–137, October 2009.

[14] H. G. Okuno, K. Nakadai, K. I. Hidai, H. Mizoguchi, and H. Kitano, "Human-robot interaction through real-time auditory and visual multiple-talker tracking," in *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 1402–1409, November 2001.

[15] I. Fasele and J. R. Movellan, "Comparison of neutrally inspired face detection algorithms," in *Proceedings of the International Conference on Artificial Neural Networks*, pp. 1395–1401, 2002.

[16] T. Tasaki, S. Matsumoto, H. Ohba et al., "Dynamic communication of humanoid robot with multiple people based on interaction distance," *Transactions of the Japanese Society for Artificial Intelligence*, vol. 20, pp. 209–219, 2005.

[17] H. Takeshima, T. Ida, and T. Kaneko, "Extracting object regions using locally Es-timated probability density functions," in *Proceedings of the Conference on Machine Vision Applications*, 2007.

[18] J. H. Ward, "Hierarchical grouping to optimize an objective function," *Journal of the American Statistical Association*, vol. 85, no. 301, pp. 236–244, 1963.

[19] T. Tasaki and F. Ozaki, "Obstacle classification and location by using a mobile omnidirectional camera based on tracked floor boundary points," in *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 5222–5227, October 2009.

[20] Y. Negishi, J. Miura, and Y. Shirai, "Calibration of omnidirectional stereo for mobile robots," in *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 2600–2605, October 2004.

[21] N. Mitsunaga, T. Miyashita, H. Ishiguro, K. Kogure, and N. Hagita, "Robovie-IV: a communication robot interacting with people daily in an office," in *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 5066–5072, October 2006.

[22] T. Kanda and H. Ishiguro, "Friendship estimation model for social robots to understand human relationships," in *Proceedings of the 13th IEEE International Workshop on Robot and Human Interactive Communication*, pp. 539–544, September 2004.

[23] J. Y. Bouguet, *Pyramidial Implementation of the Lucas Kanade Feature Tracker*, OpenCV Documentation, Intel Corporation, Microprocessor Research Labs, 1999.

[24] R. Kawanishi, A. Yamashita, and T. Kaneko, "Estimation of camera motion with feature flow model for 3D environment modeling by using omni-directional camera," in *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 3089–3094, October 2009.

[25] B. D. Lucas and T. Kanade, "An iterative image registration technique with an application to stereo vision," in *Proceedings of the International Joint Conference on Artificial Intelligence*, pp. 674–679, 1981.

[26] M. Piaggio, R. Formaro, A. Piombo, L. Sanna, and R. Zaccaria, "An optical-flow person following behavior," in *Proceedings of the IEEE ISIC/CIRNISAS Joint Conference*, pp. 4078–4083, 1998.

[27] T. Tasaki, K. Komatani, T. Ogata, and H. G. Okuno, "Spatially mapping of friendliness for human-robot interaction," in *Proceedings of the International Conference on Intelligent Robots and Systems*, pp. 1277–1282, 2005.