3-D vision-assist guidance for robots or the visually impaired

Du-Ming Tsai, Hao Hsu and Wei-Yao Chiu
Department of Industrial Engineering and Management, Yuan-Ze University, Taiwan, Republic of China

Abstract

Purpose – This study aims to propose a door detection method based on the door properties in both depth and gray-level images. It can further help blind people (or mobile robots) find the doorway to their destination.

Design/methodology/approach – The proposed method uses the hierarchical point–line region principle with majority vote to encode the surface features pixel by pixel, and then dominant scene entities line by line, and finally the prioritized scene entities in the center, left and right of the observed scene.

Findings – This approach is very robust for noise and random misclassification in pixel, line and region levels and provides sufficient information for the pathway in the front and on the left and right of a scene. The proposed robot vision-assist system can be worn by visually impaired people or mounted on mobile robots. It provides more complete information about the surrounding environment to guide safely and effectively the user to the destination.

Originality/value – In this study, the proposed robot vision scheme provides detailed configurations of the environment encountered in daily life, including stairs (up and down), curbs/steps (up and down), obstacles, overheads, potholes/gutters, hazards and accessible ground. All these scene entities detected in the environment provide the blind people (or mobile robots) more complete information for better decision-making of their own. This paper also proposes, especially, a door detection method based on the door's features in both depth and gray-level images. It can further help blind people find the doorway to their destination in an unfamiliar environment.

Keywords Vision-assist, Robot guide dog, Machine vision, Depth image, Visually impaired

Paper type Research paper

1. Introduction

Blind or visually impaired people often find it difficult to get around in new environments. The mobility aids of the blind are mainly based on long canes or guide dogs. Long canes are commonly used by the blind to detect objects within a very limited area in the path. Guide dogs give the blind more flexibility to travel in an unfamiliar environment. The main function of guide dogs is to maneuver the blind people around obstacles, indicate the location of curbs and stairs and prevent hazardous areas. Despite the advantages of guide dogs, the training time is very long and the training and maintenance costs are also very high.

Due to recent advances in computer vision and mobile robots, the robot guide dog is an attractive alternative to provide mobility aids for blind people. A robot guide dog should be equipped with the capability to detect the pathway and obstacles. An important topic in the development of robot guide dogs is robot navigation (Jacoff *et al.*, 2002; Milella *et al.*, 2008; Bonin-Font *et al.*, 2008). Robot navigation aims at the path planning of mobile robots. It generates a collision-free path in an environment with obstacles. It typically concerns the office environment that contains only walls, corridors and flat floor surfaces.

The low-cost light-weight Kinect depth sensor marketed in recent years has expanded its applications to robot navigation. Cunha *et al.* (2011) studied the robot navigation with the Kinect. The system detects walls and obstacles in an office environment. Sales *et al.* (2012) used the Kinect sensor to collect the three-dimensional (3D) point cloud of the scene. The neural network is then trained to recognize different environment configurations. Biswas and Veloso (2012) used the fast sampling plane filtering to reduce the volume of the 3D Kinect point cloud, and matched the observed 3D point cloud with a two-dimensional (2D) blueprint map.

The robot guide dog has long been investigated, since the late 1970s. Tachi *et al.* (1981) presented a robot system composed of ultrasonic, optical and landmark sensors for guiding blind people. It optically detected artificial landmarks affixed on every crossing of the road. The ultrasonic and optical sensors were used to detect obstacles on the road. Kotani *et al.* (1996) developed a robotic travel aid that was equipped with vision, sonar and tactile sensors on a motorized wheelchair. It can guide the visually impaired to avoid obstacles along a road with lane marks or a sidewalk marked with Braille. Kulyukin *et al.* (2006) developed a robot-assisted path finding system for the visually impaired. It requires the placement of radio-frequency identification (RFID) tags in the indoor environment. Galatas *et al.* (2011) presented an assistive-guide robot to avoid obstacles and to find pathways. It uses a laser ranging system for obstacle detection. Path guidance is carried out by finding two straight lines intersected at a vanished point in the image.

The current issue and full text archive of this journal is available at **www.emeraldinsight.com/0143-991X.htm**

Industrial Robot: An International Journal
41/4 (2014) 351–364

[DOI 10.1108/IR-12-2013-427]

Rodriguez *et al.* (2012) designed an obstacle detection system. A dense disparity map is computed from the images of a stereo camera. It uses the random sample consensus (RANSAC) filtering to estimate the ground plane. It then calculates the height of every pixel to the ground plane to find possible obstacles.

The currently available robot navigation and robot guide dog systems mainly focus on path finding and obstacle avoidance. They generally find limited information, such as walls, corridors and obstacles in the environment. However, the visually impaired people need more detailed information about their surroundings. This study aims at the robot vision scheme based on depth images for a robot guide dog system. The developed robot-vision system can be worn on the neck as a mobility aid to the blind or it can be attached to a mobile robot to form a guide dog robot.

In this study, the proposed robot vision scheme provides detailed configurations of the environment, including staircases (upward and downward), curbs/steps (upward and downward), obstacles, overheads, potholes/gutters, hazards and accessible ground. This paper also proposes a door detection method based on the door's features in both depth and gray-level images. It can further help the blind people find the doorway to their destination in an unfamiliar environment.

The proposed robot vision system is implemented and tested with the Xtion depth sensor, which is basically the same as the Kinect developed originally by PrimeSense. The Xtion device provides both depth and color images. It is more compact and weighs lesser than the Kinect (only 0.2 kg). The proposed robot vision algorithms in this study can be directly applied to any 3D image devices, such as laser-range finders or stereo imaging systems, for daylight and outdoor applications.

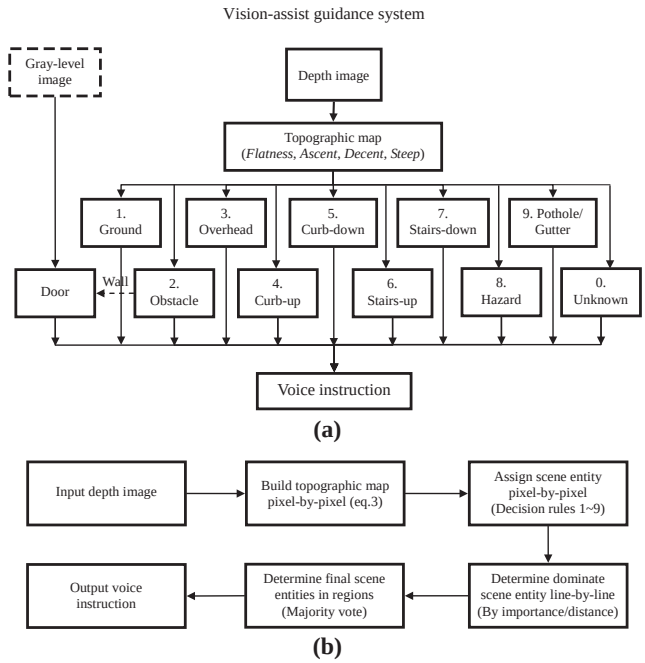
Figure 1(a) shows the configuration of the proposed vision-assist guidance system. It first converts the depth image, pixel by pixel, into a topographic map that contains four surface features, including *flatness*, *ascent*, *descent* and *steep*. The topographic map is further segmented into the regions of various scene entities in a pixel-by-pixel manner. The dominant scene entity of each line is then assigned in a line-by-line manner. Finally, the image is divided into central, left and right regions. The mode of scene entities in each region is delivered as the conclusive result in an area-by-area manner. The simplified flow diagram of the proposed algorithm is illustrated in Figure 1(b).

The paper is organized as follows. Section 2 presents the conversion from a depth image to a topographic map. The patterns and classification criteria of individual scene entities are then described. Section 3 demonstrates the detection results. The performance of the proposed system is also statistically evaluated. Section 4 concludes the paper and gives further research directions.

2. System configuration and proposed method

In this study, the proposed vision-assist guidance system is implemented with, but not limited to, an Xtion depth imaging camera. Figure 2(a) shows a person equipped with the depth-imaging system, where the Xtion device is hanging from the neck and is about at the height of the chest. The projection axis of the sensor is inclined by about 40° with respect to the ground surface, as seen in Figure 2(b).

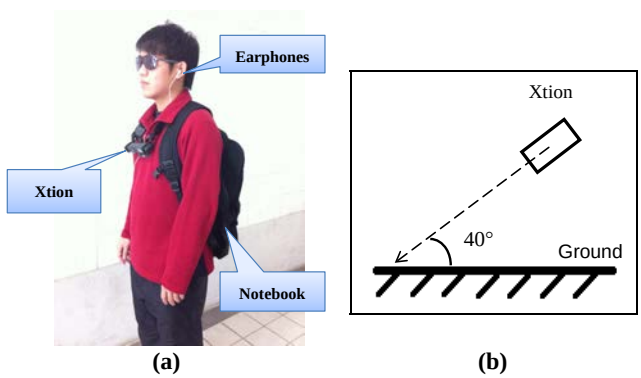
Figure 1 Configuration and flow diagram of the vision-assist guidance system



Notes: (a) Configuration of the proposed system; (b) simplified flow diagram of the proposed algorithm

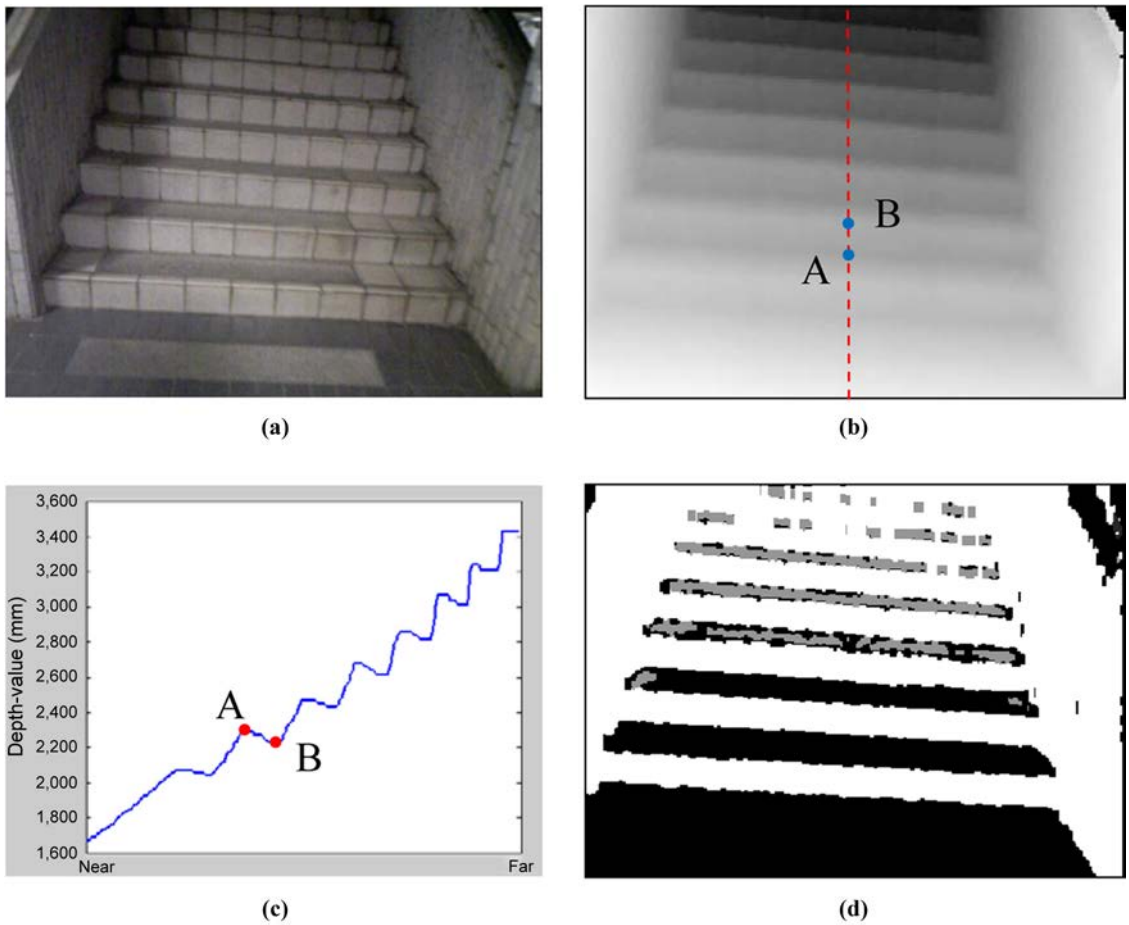
Denoted by $D(x, y)$, the depth image of size $M \times N$ from the 3D sensing device, where the depth value is in *mm* at a pixel location (x, y), for $x = 0, 1, 2 [. . .], M - 1$ and $y = 0, 1, 2 [. . .], N - 1$. Figure 3(a-c) presents the depth image and the depth profile of a vertical scan line for the upward staircase, where the ground presents steadily increasing slopes, and a step of the stairs gives an upward slope and then a downward slope in the depth profile. Figure 4(a-c) shows the depth image of the downward staircase, where a step presents an upward slope and then becomes vertically sharp. The patterns of

Figure 2 Equipment and its setting



Notes: (a) Person equipped with the Xtion depth imaging system; (b) projection axis of the depth sensor

Figure 3 Depth image and profile of an upward staircase scene



Notes: (a) Gray-level image; (b) depth image; (c) depth profile of the scan line in (b); (d) topographic map

individual scene entities in the depth profile are, thus, used for classification.

2.1 Topographic map

To identify all possible scene entities in the environment, the depth image $D(x, y)$ is first converted into a topographic map $T(x, y)$ pixel by pixel. Each pixel in the map is classified as one of the four surface features, *flatness*, *ascent*, *descent* and *steep*, based on the slope angle at the pixel in the vertical scan line of a depth profile. Let $D_x(y)$ represent the depth profile of scan line x in the depth image, i.e. $D_x(y) = D(x, y)$, $x = 0, 1, 2, \ldots, M - 1$. Because the unit of coordinates (x,y) is measured in pixels and the unit of $D_x(y)$ is given by *mm*, the depth value is normalized as

$$D'_x(y) = N \cdot [D_x(y)/\max_j D_x(j)] \quad (1)$$

where N is the image width and $\max_j D_x(j)$ is the maximum depth in scan line x. The slope angle at location y of depth profile x is given by

$$\theta(x, y) = \tan^{-1}\frac{D'_x(y + S) - D'_x(y)}{S}, \quad y = 0, 1, 2, \ldots, N - S - 1 \quad (2)$$

In this study, the support length S is set at 3. The topographic map is then given by

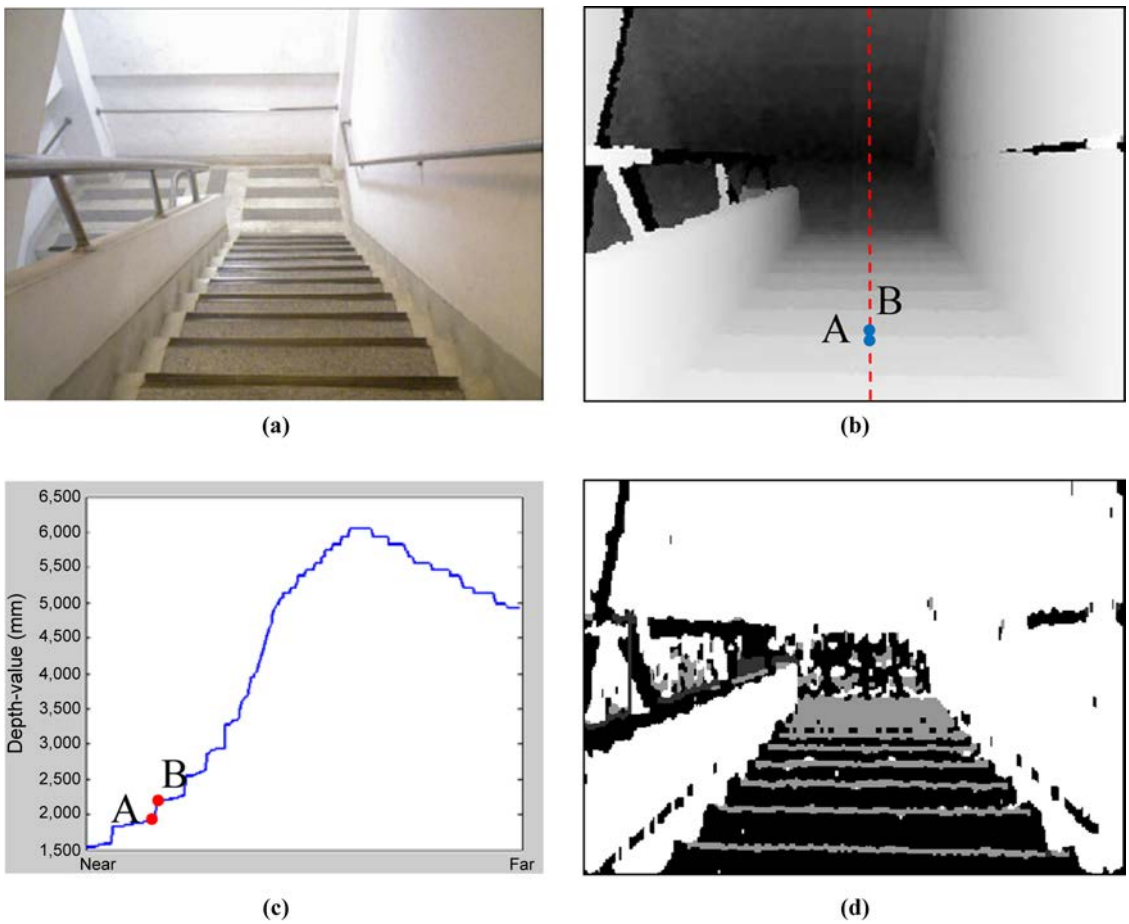
$$T(x, y) = \begin{cases} 0\ (\textit{flatness}), & \text{if } 0° \le \theta(x, y) \le 80° \\ 50\ (\textit{steep}), & \text{if } \theta(x, y) \ge 88° \\ 150\ (\textit{descent}), & \text{if } 80° < \theta(x, y) < 88° \\ 255\ (\textit{ascent}), & \text{if } \theta(x, y) < 0° \end{cases} \quad (3)$$

The numbers in the map mark the individual surface features and are used as gray values for an 8-bit visual display. Figures 3(d) and 4(d) show topographic maps of the demonstrative examples of upward-and downward-staircase entities. For the upward staircase in Figure 3, the horizontal surfaces of the stairs are marked black ($T(x, y) = 0$ for *flatness*), and the vertical surfaces of the stairs are marked white ($T(x, y) = 255$ for *ascent*).

2.2 Scene entities classification

Once the topographic map $T(x, y)$ is constructed from the depth image, it is further segmented into one of the nine scene entities based on the patterns of individual entities in a vertical scan line of the map. To do that, the run-length segments in each vertical scan line are first determined. Let $T_x(y)$ be a

Figure 4 Depth image and profile of a downward staircase scene



Notes: (a) Gray-level image; (b) depth image; (c) depth profile of the scan line in (b); (d) topographic map

vertical scan line x in the topographic map, and $\boldsymbol{c}_x^i(T_i)$ the i-th run-length with surface feature T_i in scan line x, for $T_i \in \{0$ (*flatness*), 50 (*steep*), 150 (*descent*), 255 (*ascent*)$\}$, i.e.

$$T_x(y) = T(x, y),\ x = 0, 1, 2, \ldots, M-1;$$
$$\boldsymbol{c}_x^i(T_i) = \{y_{i,s}, \ldots, y_{i,e} \mid T_x(y) = T_i\ \forall\ y = y_{i,s}, y_{i,s+1}, \ldots, y_{i,e-1}, y_{i,e}\} \quad (4)$$

where $y_{i,s}$ and $y_{i,e}$ are, respectively, the start and end points of the run. All points between $y_{i,s}$ and $y_{i,e}$ have the same surface feature. $\boldsymbol{c}_x^i(T_i)$ and $\boldsymbol{c}_x^{i+1}(T_{i+1})$ are two adjacent run-length segments with $y_{i+1,s} = y_{i,e} + 1$. The length of the run $\boldsymbol{c}_x^i(T_i)$ in pixels is given by

$$L[\boldsymbol{c}_x^i(T_i)] = y_{i,e} - y_{i,s} \quad (5)$$

It can also be converted to *mm* using the transformation in equation (6). A run-length ≤ 3, i.e. $L[\boldsymbol{c}_x^i(T_i)] \leq 3$, is treated as noise and is merged to its adjacent neighbor with a larger length. Figure 5 demonstrates the run-lengths of *flatness* (black regions with $T = 0$) and *ascent* (white regions with $T = 255$) for a vertical scan line in the topographic map of the upward staircase.

In this study, the sensed scene is categorized into nine scene entities that help the blind people (or mobile robots) understand better their surroundings. The names and code numbers of individual scene entities are listed in Figure 1. The decision rules for individual scene entities and the conversion from the topographic map $T(x, y)$ to the entity map $E(x, y)$ are described in detail as follows.

2.2.1 Ground

Ground is the accessible area to the blind. The ground or floor shows the flatness on the surface. Thus, for run-length $\boldsymbol{c}_x^i(T_i)$ with $T_i = flatness$, $E(x, y) = 1$ (ground) for all $y \in \boldsymbol{c}_x^i(T_i)$. See Figure 6(a) for the demonstration of a ground surface.

2.2.2 Obstacles

An obstacle shows the ascent feature and lasts for a significant segment in the y-axis. An obstacle example is demonstrated in Figure 6(b). For a run-length with *ascent* feature, an obstacle code is assigned if the length is $>$ 20 per cent of the image width N in the y-axis, i.e. if $L[\boldsymbol{c}_x^i(T = ascent)]/N > 0.2$, then $E(x, y) = 2$ (obstacles) for all $y \in \boldsymbol{c}_x^i(T = ascent)$.

2.2.3 Overhead

An object overhung from above is also a type of obstacle, but does not touch the flat ground. The depth profile shows first a segment of ground, and then suddenly drops with a significant distance and becomes an ascent segment. To calculate the height of the overhead from the ground, the two

Figure 5 Topographic map and run lengths of an upward staircase scene

Notes: (a) Gray-level image; (b) depth image; (c) topographic map, where black regions are flat surfaces and white regions are ascending surfaces of steps

extreme points P_1 (the end point of the *flatness* run-length) and P_2 (the start point of the *ascent* run-length) of the drop are first identified. Let point P_2' be the horizontal intersection of P_2 and the ground segment in the depth profile. It gives the vertical projected point of P_2 on the ground. The distance between P_2 and P_2' is then the height of the overhead above the ground. Figure 6(c) shows the demonstrative example of an overhead object.

Figure 6 Profile patterns of individual scene entities

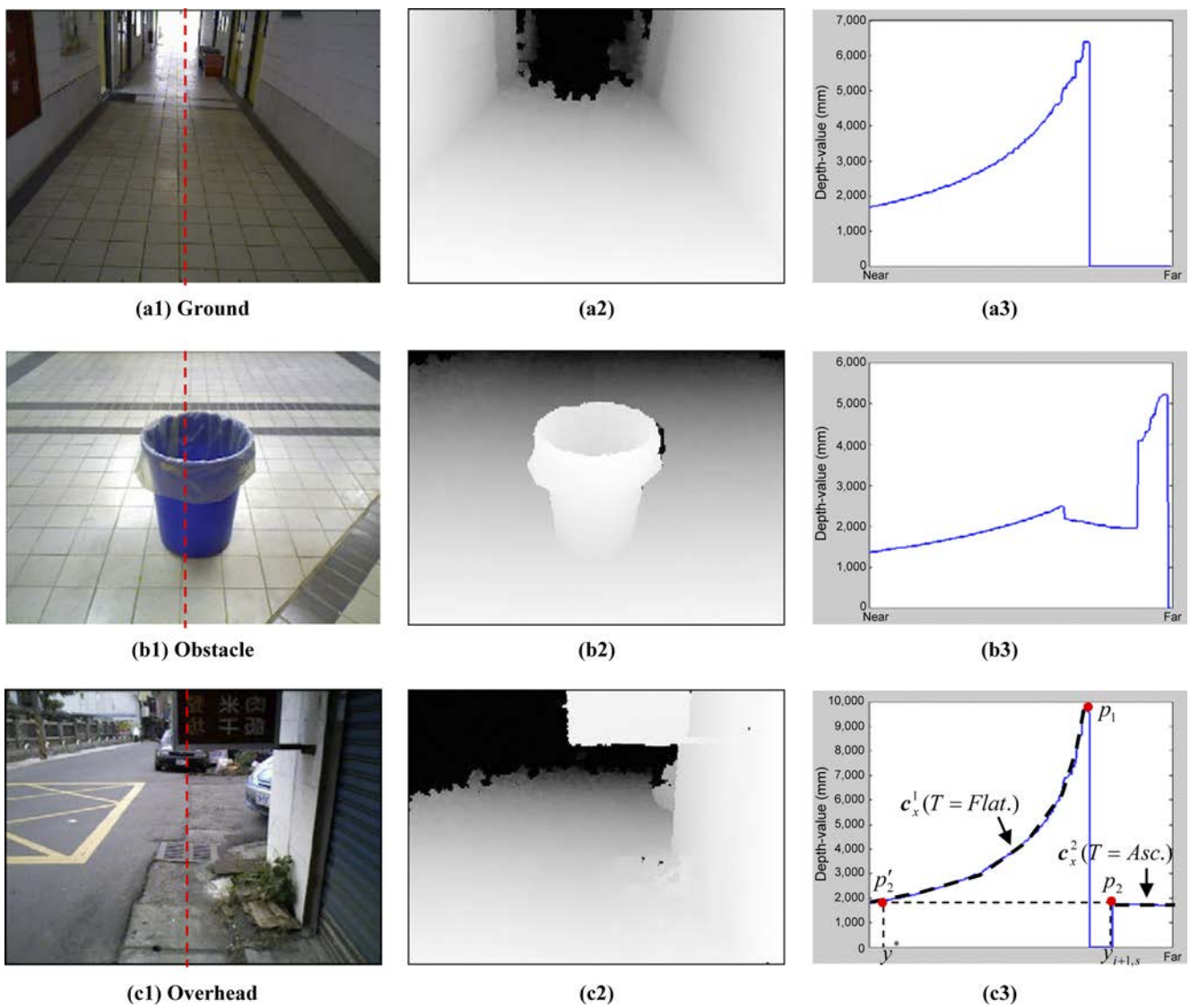
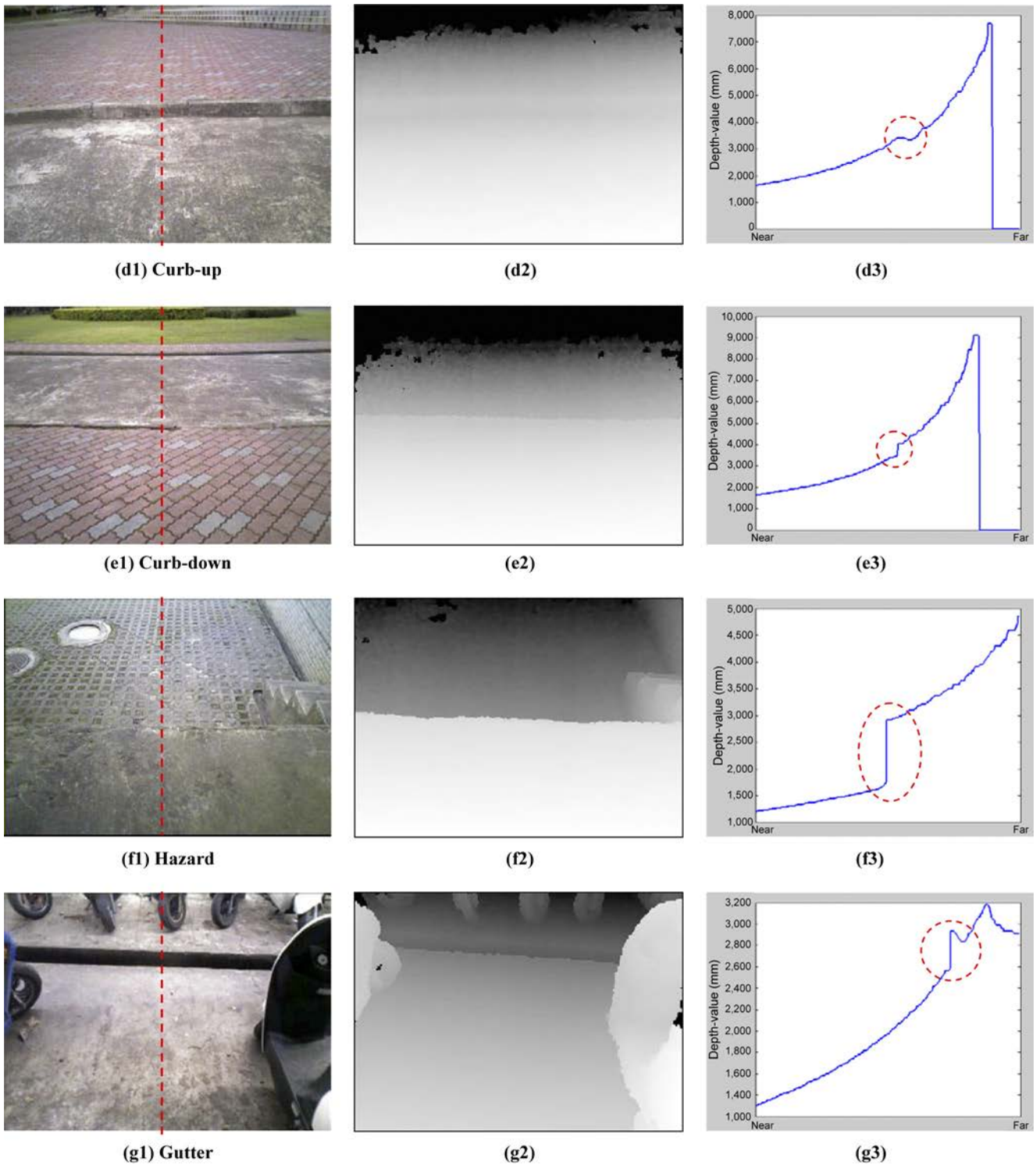


Figure 6 Continued



Notes: (a1)-(a3) Gray-level image, depth image and a depth profile of the ground entity; (b1)-(b3) obstacle; (c1)-(c3) overhead and its height calculation; (d1)-(d3) up-the-curb; (e1)-(e3) down-the-curb; (f1)-(f3) hazardous dock; (g1)-(g3) gutter

For two adjacent run-lengths $\boldsymbol{c}_x^i(T = flatness)$ and $\boldsymbol{c}_x^{i+1}(T = ascent)$ with

$$\overline{P_1P_2} = D(x, y_{i,e}) - D(x, y_{i+1,s}) > 100 \text{ mm},$$

If the distance between $P_2 = (x, y_{i+1,s}, D(x, y_{i+1,s}))$ and $P_2' = (x, y^*, D(x, y^*))$ is larger than a height threshold T_h, then $E(x, y) = 2$ (overhead), $\forall\, y \in \boldsymbol{c}_x^{i+1}(T = ascent)$, where $y^* \in \boldsymbol{c}_x^i(T = flatness)$ and $D(x, y^*) = D(x, y_{i+1,s})$. The height in *mm* can be calculated from equation (6), and the threshold T_h is given by 1.0 *m* in this study. If the height constraint T_h is violated, the object is classified as an obstacle.

2.2.4 Curb/step-up

Up the curb or step represents a small ascent segment with large flatness segments on both sides of the ascent segment. Hence, given three consecutive run lengths $\boldsymbol{c}_x^i(T_i)$, $\in \boldsymbol{c}_x^{i+1}(T_{i+1})$ and $\boldsymbol{c}_x^{i+2}(T_{i+2})$, if $T_i = flatness$, $T_{i+1} = ascent$, $T_{i+1} = flatness$ and $S_{min} < L[\boldsymbol{c}_x^{i+1}(T = ascent)] < S_{max}$,

$$L[\boldsymbol{c}_x^i(T = flatness)], L[\boldsymbol{c}_x^{i+2}(T = flatness)] > S_{max},$$

where S_{min} and S_{max} are the minimum and maximum widths of a step (or curb), then $E(x,y) = 4 \;\; \forall\, y \in \boldsymbol{c}_x^{i+1}(T = ascent)$. Figure 6(d) shows the depth image and a line profile of an up-the-curb example.

2.2.5 Curb/step-down

Down the curb or step involves a small descent segment with larger flatness segments on both sides of the descent segments. The decision rule is the same as that for the curb/step-up above by replacing $\boldsymbol{c}_x^{i+1}(T = ascent)$ with $\boldsymbol{c}_x^{i+1}(T = descent)$ and $E(x, y) = 5$, $\forall\, y \in \boldsymbol{c}_x^{i+1}(T = descent)$. Figure 6(e) presents an example of the down-the-curb case.

2.2.6 Upward staircase

The upward-staircase entity shows the repetitive interchange of an ascent and flatness segment, where each segment is about the width of a step. Hence, for adjacent run lengths $\boldsymbol{c}_x^{i+K}(T_{i+K})$ and $\boldsymbol{c}_x^{i+1+K}(T_{i+1+K})$ with $K = 0$ and 2, if $T_{i+K} = ascent$ and $T_{i+1+K} = flatness$ and $S_{min} < L[\boldsymbol{c}_x^{i+K}(T = ascent)]$, $L[\boldsymbol{c}_x^{i+1+K}(T = flatness)] < S_{max}$, then $E(x,y) = 6$, $\forall\, y \in \boldsymbol{c}_x^{i+K}(T = ascent) \cup \boldsymbol{c}_x^{i+1+K}(T = flatness)$.

The decision rule above requires at least two consecutive steps ($K = 0$ and 2) to form a staircase. The demonstrative example of an upward-staircase entity can be seen in Figure 3.

2.2.7 Downward staircase

The downward-staircase entity shows the repetitive interchange of a descent and flatness segment, where each segment is about the width of a step. Figure 4 depicts the depth image and a line profile of the downward staircase. The decision rule requires at least 2 downward steps, and is given by: for adjacent run lengths $\boldsymbol{c}_x^{i+K}(T_{i+K})$ and $\boldsymbol{c}_x^{i+1+K}(T_{i+1+K})$ with $K = 0$ and 2, if $T_{i+K} = descent$ and $T_{i+1+K} = flatness$ and $S_{min} < L[\boldsymbol{c}_x^{i+K}(T = descent)]$, $L[\boldsymbol{c}_x^{i+1+K}(T = flatness)] < S_{max}$, then $E(x,y) = 7$, $\forall\, y \in \boldsymbol{c}_x^{i+K}(T = descent) \cup \boldsymbol{c}_x^{i+1+K}(T = flatness)$.

2.2.8 Hazard

A hazard involves a steep segment of significant drop, such as the platform edges at a railway or subway station. Figure 6(f) shows a line profile from the depth image of a truck loading dock. The decision rule can be simply given by:

if $L[\boldsymbol{c}_x^i(T = steep)] > S_{max}$, then

$$E(x,y) = 8, \forall\, y \in \boldsymbol{c}_x^i(T = steep)$$

2.2.9 Potholes/gutters

A hole or gutter shows a cavity in the ground. The depth profile of a pothole or gutter in the pavement presents sequentially flatness segment, descent segment, ascent segment and then back to flatness segment. Figure 6(g) gives the demonstrative example of a gutter in the motorcycle parking lot. The decision rule is thus given by: for four consecutive run lengths $\boldsymbol{c}_x^i(T_i)$, $\boldsymbol{c}_x^{i+1}(T_{i+1})$, $\boldsymbol{c}_x^{i+2}(T_{i+2})$, and $\boldsymbol{c}_x^{i+3}(T_{i+3})$, if $T_i = flatness$, $T_{i+1} = descent$, $T_{i+2} = ascent$ and $T_{i+3} = flatness$, and $L[\boldsymbol{c}_x^{i+1}(T = descent)]$, $L[\boldsymbol{c}_x^{i+2}(T = ascent)] > S_{min}$, then $E(x,y) = 9$, $\forall\, y \in \boldsymbol{c}_x^{i+1}(T = descent) \cup \boldsymbol{c}_x^{i+2}(T = ascent)$.

2.3 Dominant scene entity

The scene entities in $E(x, y)$ are assigned to pixels based on individual vertical scan lines. Given a vertical scan line x, it may contain more than one scene entity. Only the most important entity in the line is retained for further consideration. The assignment of the dominant scene entity in a scan line is based on the following two criteria:

1. *Importance:* The scene entities that could be dangerous to the user have the higher priority. The hazard entity has the top priority. The remaining sequences are potholes/gutters, overheads, obstacles, downward staircase, curb/step-down, upward staircase, curb/step-up and ground.
2. *Distance:* The subject closer to the user has a higher priority.

The dominant scene entity in a scan line x is denoted by $G(x)$, where $G(x) \in \{0, 1, 2\ [\ldots], 9\}$, for $x = 0, 1, 2\ [\ldots], M - 1$. The code numbers represent the scene entities given in Figure 1. Figure 7 shows the gray-level images of upward and downward staircases in (a1)-(b1); the corresponding topographic maps $T(x, y)$ in (a2)-(b2), the scene entity maps $E(x, y)$ in (a3)-(b3); and the resulting dominant entities $G(x)$ of individual scan lines in (a4)-(b4).

Finally, the scene image is vertically divided into three regions, R_l, R_c and R_r, representing the left, center and right of a scene. The ratio of these three regions is given by 1:4:1, as seen in Figure 8. The central region R_c takes two-third of the scene image that represents the accessible area of a long cane in front of a blind person. The additional information in the left and right of the scene provides helpful instruction to guide the user to access his/her destination more efficiently and effectively. The final resulting entity in each of the three divided regions is determined by the majority-vote rule, i.e.

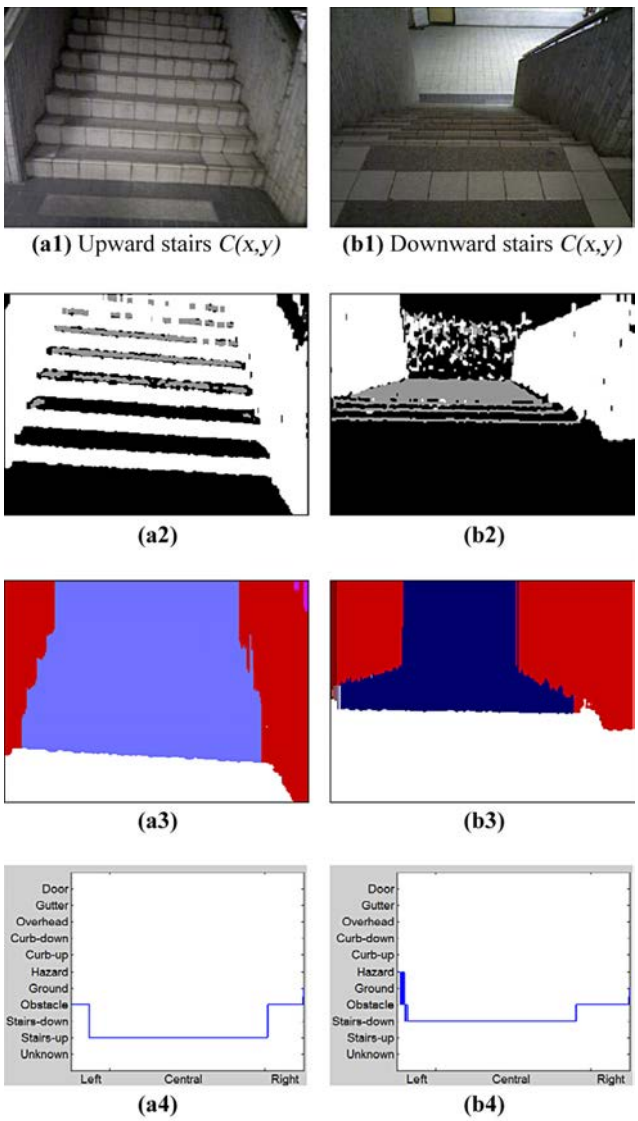
$$G_l = \text{mode}\ \{G(x), \forall x \in R_l\},$$

$$G_c = \text{mode}\ \{G(x), \forall x \in R_c\},$$

$$G_r = \text{mode}\ \{G(x), \forall x \in R_r\}.$$

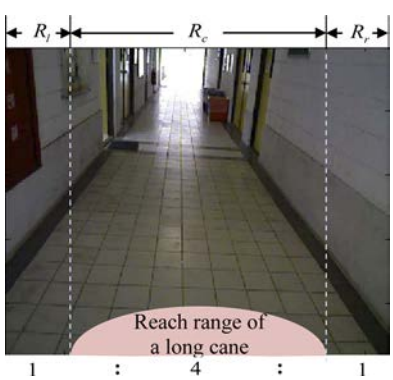
The pixel–line region assignment of scene entities with the majority vote criterion gives a robust vision-assist guidance system that can tolerate noise and random misclassification of surface features and scene entities.

Figure 7 Assignment of dominant scene entities



Notes: (a1)-(b1) gray level images of staircases; (a2)-(b2) respective topographic maps $T(x,y)$; (a3)-(b3) respective entity maps $E(x,y)$; (a4)-(b4) resulting dominat entities $G(x)$

Figure 8 Left, central and right partitioned regions of a scene image



2.4 Door detection

Doors are significant landmarks for finding the pathway. Vision-based door detection methods have been proposed by analyzing the color, texture and geometric features of doors. Lee *et al.* (2004) used door width, doorframes, difference of depth and doorknob as door features. A Bayesian network is then trained for door identification. Chen and Birchfield (2008) used color, texture and intensity edges in color images as door features. The Adaboost algorithm is then applied for classification. Hensler *et al.* (2009) further used color, doorknob, door gap, door frame and texture extracted from color images and also door width and door concavity obtained from laser-range images to characterize doors. They then also used the Adaboost algorithm for classification. Mahmood and Kuwar (2012) used door width, door concavity and bottom-door gap as features. An unsupervised self-organizing map is applied for door identification. The currently available methods for door detection in the literature can be used as a supplement module of the vision-assist guidance system.

Many of the door detection algorithms in the current literature require reliable low-level line and/or corner detection process to extract door features, and require a training process to construct the classifier from a huge number of training samples. They generally result in relatively low processing rate. For example, the door detection algorithm proposed by Hensler *et al.* (2009) gives a processing rate of 12 fps. It may be fast enough for just door detection. However, door detection is only one of the functional modules for the robot-vision guidance system. They may not meet the time requirement when all scene entities and doors must be detected simultaneously.

In this study, we use the cues from both the depth and gray-level images to identify doors. To detect a door, a wall must be first recognized. A wall presents the change from a ground entity to an obstacle entity in the scene. There is a clear straight line that separates the ground and the adjacent obstacle. If the condition above is satisfied, the detected obstacle in section 2.3 is identified as a wall. A door on the wall must contain the following three properties:

1. A door frame consists of two straight lines in the vertical direction, and these two straight lines in the wall region must connect to the flat ground in the gray-level image.
2. A door is generally intruded on the wall, i.e. it is not on the same surface of the wall. The two vertical straight lines of the door frame must show a small step change in depth in the depth image.
3. The width of a door must follow the convention.

To verify the three properties of a door, the region of a detected wall in the depth image $D(x, y)$ is mapped to its corresponding coincidence in the gray-level image. The Sobel edge detector is then used to find edge points in the wall region in the gray-level image. Only the gradient operator in the x direction is required to detect the vertical edges of the door frame. Denote by L_{wall} the line equation of the intersection line between the ground and the wall surface in the gray-level image. The vertical lines of a door frame may not be perpendicular to the line L_{wall} in the gray-level image due to the viewing angle of the imaging sensor. To find efficiently the vertical lines of a door frame, the vertical projection of edge points (i.e. the number of edge points for each vertical line is accumulated) in the Sobel-edge image is only performed within a narrow strip region defined by two parallel lines L_{wall} and

$L_{wall} + \Delta L$, where ΔL is the width of the strip. The short line segments in the narrow strip can be considered vertical in the gray-level image. Figure 9 depicts the required details for door detection. When the projected values are significantly large, the potential vertical edges of the door frame are present in the scene. Principal component analysis (PCA) is applied to the edge points to estimate the line equation. When the total number of edge points in the estimated line is large enough and the edge points in the line show step changes with respect to their horizontal neighborhood in the depth image, the door requirements of first and second properties above are satisfied.

Finally, to find the door width, let $P_s = (x_s, y_s, d_s)$ and $P_e = (x_e, y_e, d_e)$ be the points of two vertical straight lines connected to the ground (refer to Figure 9(e) for the details). Since the coordinates (x_s, y_s) are measured in pixels and d_s in millimeters in the depth image, they are all converted into millimeters as follows (NQT, 2013):

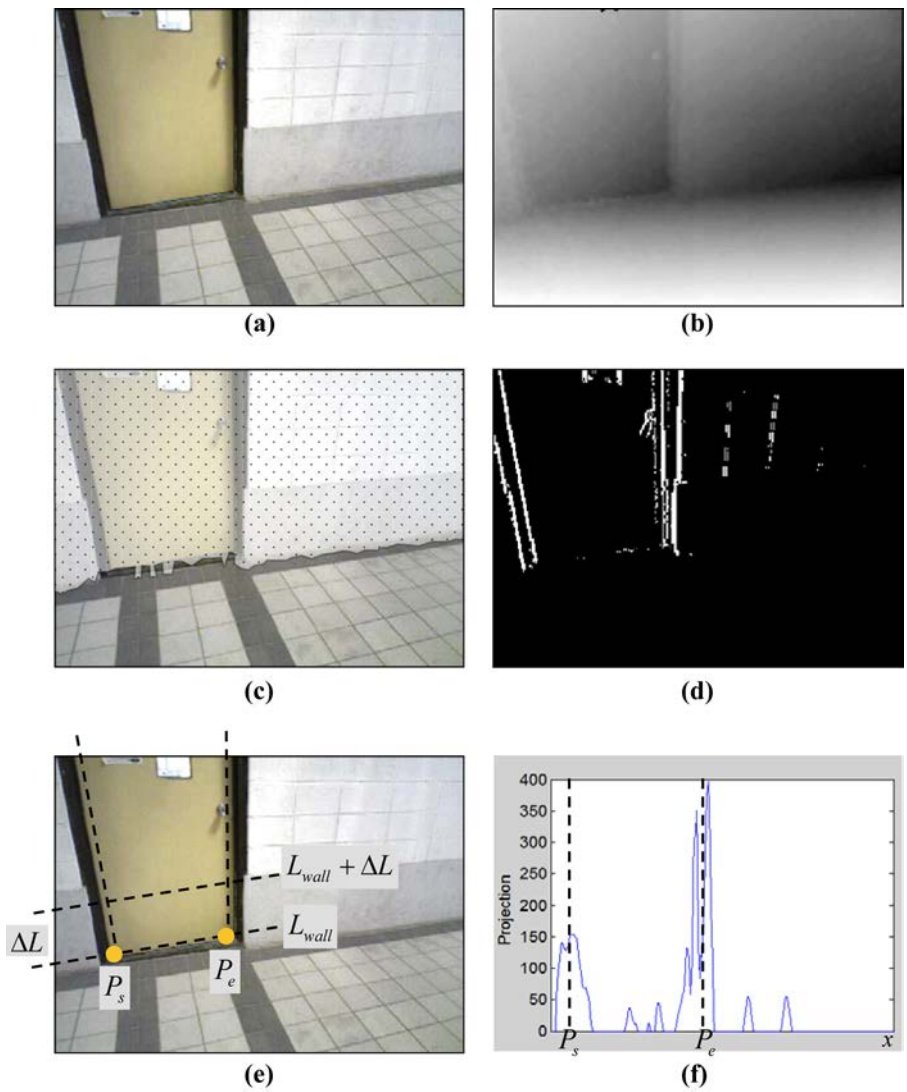
$$\begin{aligned} x_s' &= d_s \cdot (x_s - M/2) \cdot \tan(\alpha/2)/(M/2) \\ Y_s' &= d_s \cdot (y_s - N/2) \cdot \tan(\beta/2)/(N/2) \end{aligned} \quad (6)$$

where the depth image is of size $M \times N$ pixels, and α and β are the horizontal and vertical angular views, respectively. Likewise, the point coordinates (x_e, y_e) are also converted using equation (6). The door width is then the distance between $P_s' = (x_s', y_s', d_s)$ and $P_e' = (x_e', y_e', d_e)$, i.e.

$$W = \sqrt{(x_s' - x_e')^2 + (y_s' - y_e')^2 + (d_s - d_e)^2}$$

If W is larger than the width of a door in convention, the third requirement of a door is satisfied. The proposed door detection

Figure 9 Door detection from gray-level and depth images



Notes: (a) Gray-level image; (b) depth image; (c) gray-level image and the detected wall; (d) vertical edges detected in the wall region of the gray-level image; (e) calculation of door features; (f) vertical projection of pixel points in the strip bounded by L_{wall} and $L_{wall} + \Delta L$

scheme uses simple IF–THEN rules with door features in both color and depth images. It requires no training process, and can achieve a very fast processing rate of 50 fps for door detection.

3. Tests and results

As mentioned previously in section 1, the proposed method for vision-guided assistance is implemented with the depth imaging device Xtion. All algorithms developed in this paper are coded in the C++ language, and are executed on a notebook with an Intel Core2 Duo 2, 53GHz processor. For images of size 320 × 240 pixels, the processing rate is 23 fps without enabling the door detection. It is then 15 fps with all functions in action. The computation time is fast enough for real-time guidance. Table I summarizes the execution time for each detailed task in the vision-assist guidance system. It takes only 0.0438 seconds/frame for scene entity identification and 0.0198 seconds/frame for door detection.

Table I Computation time for each task in the vision-assist guidance system

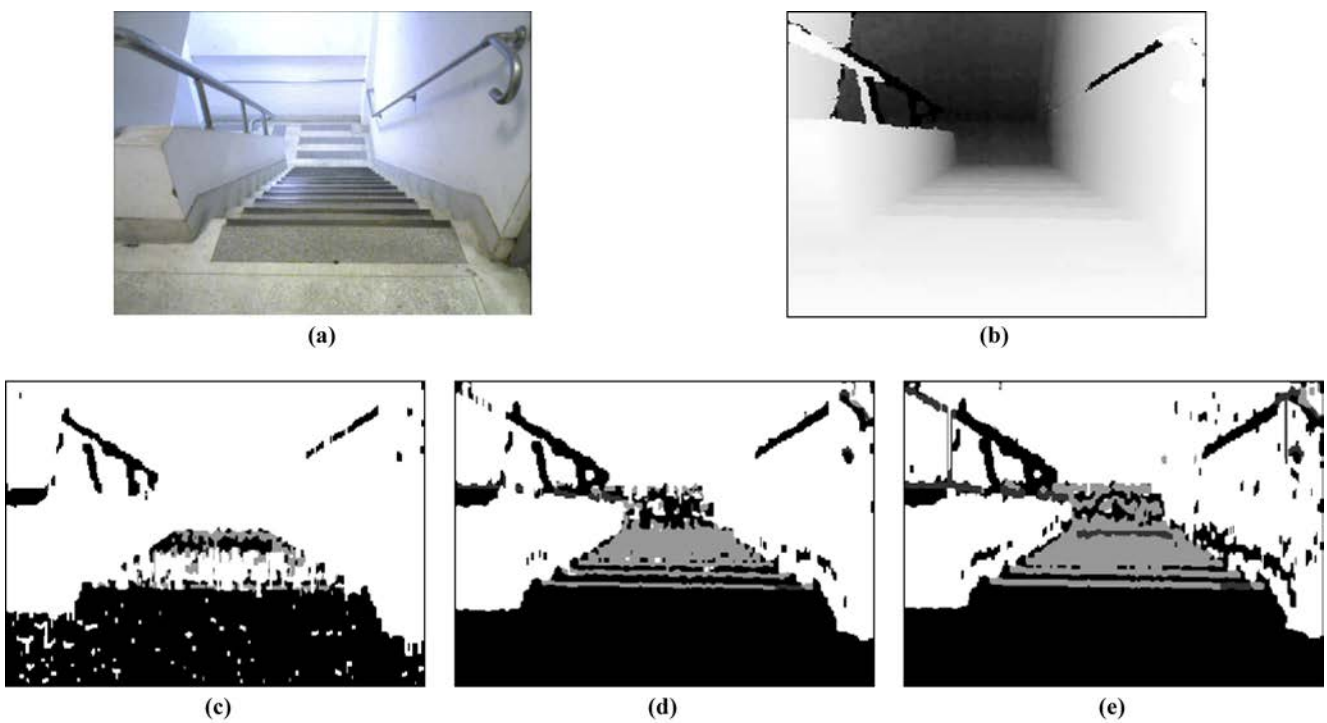
Task	Time (Seconds)
Scene entities	
Topographic map	0.0354
Scene entity map	0.0068
Dominant entities in scan lines	0.0010
Prioritized entities in regions	0.0006
Sum	0.0438 (23 fps)
Door detection	0.0198
Overall time	0.0636 (15 fps)

The line-and region-based decisions on scene entities can tolerate noise well. We thus need only evaluate the effect of support length S for inclined angle calculation in equation (2). Figure 10 presents a downward staircase scene and the corresponding topographic maps with varying support length S. It shows that a very small support length $S = 1$ yields severe noise points. Support lengths ≥ 3 generate similar results. The support length $S = 3$ is used in this study. The 3 × 3 majority filter is applied to the topographic map to remove random noisy points.

Figure 11 shows the upward staircase on the left-hand side of the image. It reveals that the proposed method can still reliably detect the partial stairs in the left image region R_l. Note that the exit in the right of the scene is also well detected as an accessible ground in the right image region R_r. This gives sufficient information to the blind people (or mobile robot) for better decision-making. Figure 12 presents the downward staircase, where the staircase entity is not straightly in front of the user. The proposed system can still reliably identify the downward staircase in the central image region R_c. Note that the upward staircase in the left image region is also correctly detected. Figure 13 demonstrates the detection results of various doors in different scene environments. The proposed method has shown its capability for the identification of doors with different structural patterns on the door surfaces.

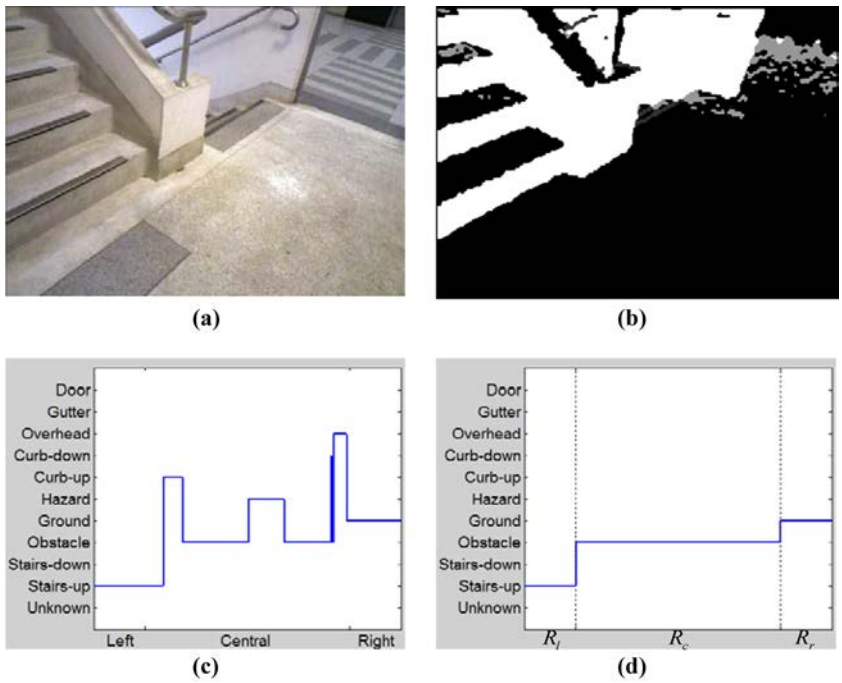
To further evaluate the performance of the proposed vision-assist guidance system, we have collected a large number of scene samples for each individual scene entity from various environments, including campus, streets, subway

Figure 10 Effect of varying S value on topographic maps



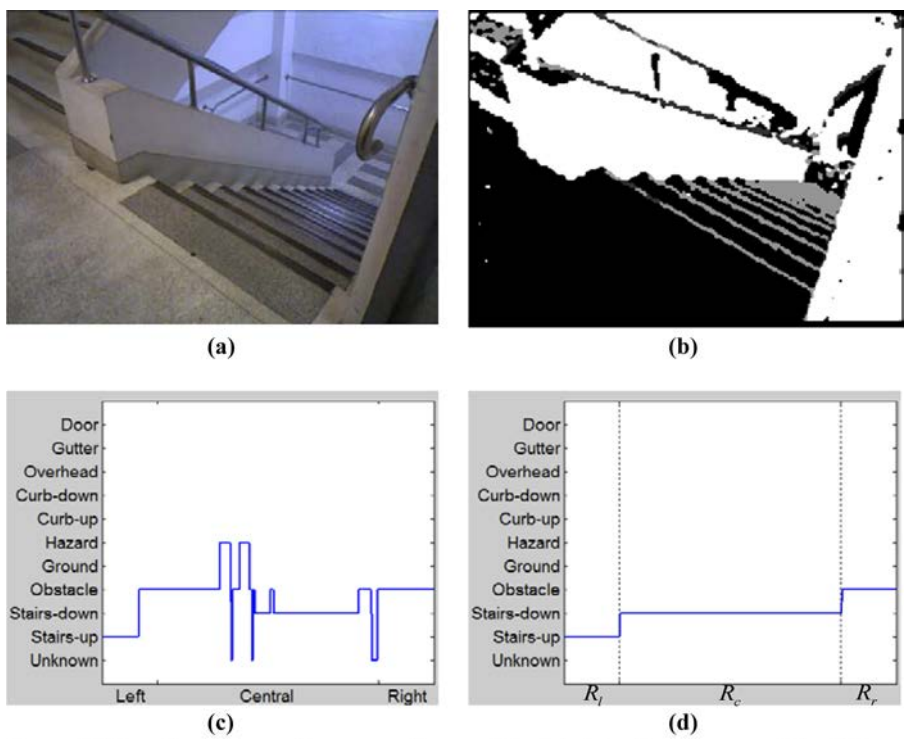
Notes: (a) Gray-level image of a downward staircase scene; (b) depth image; (c)-(e) resulting topographic maps with $S = 1$, 3 and 5, respectively

Figure 11 Detection result for a scene with accessible ground and partial upward staircases



Notes: (a) Gray-level image; (b) topographic map $T(x,y)$; (c) dominant scene entities $G(x)$; (d) resulting scene entities in three regions $R_l - R_c - R_r$

Figure 12 Detection results for a scene with downward and upward staircases



Notes: (a) Gray-level image; (b) topographic map $T(x,y)$; (c) dominant scene entities $G(x)$; (d) resulting scene entities in three regions

Figure 13 Detection results of doors

Notes: (a1)-(c1) doors of different surface structures; (a2)-(c2) corresponding detection results

station and supermarket. Table II lists the numbers of individual scene entities collected from four different environments. The test samples for each scene entity type is generally more than one hundred. In addition, there are 99 door scenes collected from campus and apartment environments for the experiment. The confusion matrix shown in Table III reveals that the proposed system can reliably detect all scene entities from the 3,150 test scene samples. The proposed door detection scheme also achieves a high detection rate. Only six door scenes out of 99 door samples are misclassified as obstacles. The errors are simply due to indistinguishable colors between a door and its surrounding wall in the color image.

The proposed vision-assist guidance system has also been tested by a born-blind person. She has positively indicated

Table II Collected samples of individual scene entities in different environments

Scene entity	Campus	Streets	Subway station	Supermarket	Total
Ground	529	399	299	348	1575
Obstacle	93	79	72	78	322
Overhead	52	44	–	–	96
Curb-up	67	53	45	47	212
Curb-down	66	52	44	46	208
Stairs-up	65	43	48	62	218
Stairs-down	67	44	48	63	222
Hazard	62	43	42	52	199
Pothole/gutter	57	41	–	–	98

Table III Confusion matrix of scene entities and doors with the proposed method

	Ground	Obstacle	Overhead	Curb-up	Curb-down	Stair-up	Stair-down	Hazard	Pothole/gutter	Door
Ground	1575	0	0	0	0	0	0	0	0	0
Obstacle	0	322	0	0	0	0	0	0	0	0
Overhead	0	0	96	0	0	0	0	0	0	0
Curb-up	0	0	0	212	0	0	0	0	0	0
Curb-down	0	0	0	0	208	0	0	0	0	0
Stair-up	0	0	0	0	0	218	0	0	0	0
Stair-down	0	0	0	0	0	0	222	0	0	0
Hazard	0	0	0	0	0	0	0	199	0	0
Pothole/gutter	0	0	0	0	0	0	0	0	98	0
Door	0	6	0	0	0	0	0	0	0	93

Notes: In the confusion matrix, the dark-gray cells highlight the correctly identified entity samples. The light-gray cell presents the misclassified entity samples

that the system can promptly respond the scene entity from a longer distance, compared to the search distance of a long cane. This gives the blind people more time to respond to changes in their surroundings. She has also indicated that the system not only provides the scene information in front of the user but also gives the useful scene information on the left and right of the user. It helps the blind people make better decisions to avoid obstacle/hazard zones and choose the right move to their destination.

4. Conclusions

In this paper, we have presented a vision-assist guidance system that can provide detailed scene information of the surrounding environment. The detailed results can be used by blind people or mobile robots for safer and more effective guidance in their pathway to the destination. The depth-imaging-based system can currently detect accessible ground, upward staircase, downward staircase, upward curb/step, downward curb/step, obstacles, overheads, potholes/gutters, hazard and doors. It applies the point-line-region principle with majority vote to identify scene entities. This approach is very robust to noise and random misclassification. The proposed machine vision scheme is implemented with a light-weight depth camera, and evaluated by a naive blind person. It can be utilized with any other 3D imaging devices such as laser-range finder and stereo imaging equipment mounted on mobile robots.

The proposed pixel–line region encoding of the scene gives the flexibility for setting the decision rules of new additional scene entities, such as detection of corners for L-or T-pathways. In the future, the proposed vision-assist guidance system can be incorporated with smart phones or global positioning system (GPS) devices. The smartphone first recognizes the voices from the visually impaired. It then searches for destination and provides the route to the user by using the Google map service. While the visually impaired follow the instructed route to the destination, the proposed vision-assist guidance system then provides the scene entity information of the environment so that the user can effectively and safely walk along the route.

References

Biswas, J. and Veloso, M. (2012), "Depth camera based indoor mobile localization and navigation", *IEEE International Conference on Robotics and Automation*, *Saint Paul, MN*.

Bonin-Font, F., Ortiz, A. and Oliver, G. (2008), "Visual navigation for mobile robots: a survey", *Journal of Intelligent and Robotic Systems*, Vol. 53 No. 3, pp. 263-296.

Chen, Z. and Birchfield, S.T. (2008), "Visual detection of lintel-occluded doors from a single image", *IEEE Computer Society Workshop on Visual Localization for Mobile Platforms*, Vol. 1, pp. 1-8.

Cunha, J., Pedrosa, E., Cruz, C., Neves, A.J.R. and Lau, N. (2011), "Using a depth camera for indoor robot localization and navigation", *Robotics Science and Systems 2011 Workshop on Advanced Reasoning with Depth Cameras*, Los Angles, CA.

Galatas, G., McMurrough, C., Mariottini, G.L. and Makedon, F. (2011), "EyeDog: an assistive-guide robot for the visually impaired", *Proceedings of the 4th International Conference on PErvasive Technologies Related to Assistive Environments*, *Crete, Greece*.

Hensler, J., Blaich, M. and Bittel, O. (2009), "Real-time door detection based on AdaBoost learning algorithm", *Eurobot Conference of Communications in Computer and Information Science*, Vol. 82, pp. 61-73.

Jacoff, A., Messina, E. and Evans, J. (2002), "Performance evaluation of autonomous mobile robots", *Industrial Robot*, Vol. 29 No. 3, pp. 259-267.

Kotani, S., Mori, H. and Kiyohiro, N. (1996), "Development of the robotic travel aid HITOMI", *Robotics and Autonomous System*, Vol. 17 Nos. 1/2, pp. 119-128.

Kulyukin, V., Gharpure, C., Nicholson, J. and Osborne, G. (2006), "Robot-assisted wayfinding for the visually impaired in structured indoor environment", *Autonomous Robots*, Vol. 21 No. 1, pp. 29-41.

Lee, J.-S., Doh, N.L., Chung, W.K., You, B.-J. and Youm, Y.I. (2004), "Door detection algorithm of mobile robot in hallway using PC-camera", *International Symposium on Automation and Robotics in Construction*, pp. 456-460.

Mahmood, F. and Kuwar, F. (2012), "A self-organizing neural scheme for door detection in different environments",

International Journal of Computer Applications, Vol. 60 No. 9, pp. 13-18.

Milella, A., Cicirelli, G. and Distante, A. (2008), "RFID-assisted mobile robot system for mapping and surveillance of indoor environments", *Industrial Robot*, Vol. 35 No. 2, pp. 143-152.

NQT (2013), "Measuring using Kinect", available at: tnq177.wordpress.com/2013/02/16/metric-measuring-using-kinect/ (accessed 24 August 2013).

Rodriguez, A., Yebes, J.J., Alcantarilla, P.F., Bergasa, L.M., Almazan, J. and Cela, A. (2012), "Assisting the visually impaired: obstacle detection and warning system by acoustic feedback", *Sensors*, Vol. 12 No. 12, pp. 17476-17496.

Sales, D., Correa, D., Osorio, F.S. and Wolf, D.F. (2012), "3D vision-based autonomous navigation system using ANN and Kinect sensor", *Communications in Computer and Information Science*, Vol. 311, pp. 305-314.

Tachi, S., Tanie, K., Komoriya, K., Hosoda, Y. and Abe, M. (1981), "Guide dog robot – its basic plan and some experiments with MELDOG MARK I", *Mechanism and Machine Theory*, Vol. 16 No. 1, pp. 21-29.

Corresponding author

Du-Ming Tsai can be contacted at: iedmtsai@saturn.yzu.edu.tw