Applied Bionics and Biomechanics 10 (2013) 139–157
DOI 10.3233/ABB-140081
IOS Press

Geometric fuzzy techniques for guidance of visually impaired people

Adán Landa-Hernández, Heriberto Casarubias-Vargas and Eduardo Bayro-Corrochano*
Department of Electrical Engineering and Computer Science, CINVESTAV Campus Guadalajara Jalisco, México

Abstract. In this paper we present the design of a device to guide the visually impaired person who normally uses a cane. We propose a non-invasive device that will help blind and visually impaired people to navigate. The system uses stereoscopic vision, a RGB-D sensor and an IMU to process images and to compute the distances from obstacles relative to cameras and to search for free walking paths in the scene. This computing is done using stereo vision, vanishing points, and fuzzy rules. Vanishing points are used to obtain a main orientation in structured spaces. Since the guidance system is related to a spatial reference system, the vanishing point is used like a virtual compass that helps the blind to orient him- or herself towards a goal. Reinforced with fuzzy decision rules, the system supports the blind in avoiding obstacles, thus the blind person is able to cross structured spaces and avoid obstacles without the need for a cane.

1. Introduction

The main categories of devices to assist blind persons are: Electronic Travel Aids (ETAs), Electronic Orientation Aids (EOAs), and Position Locator Devices (PLDs). This work comes in the category of ETAs and it consists of an non-invasive device for the guidance for the blind.

Navigation by blind and visually impaired people is a subject that has not been well investigated and at the present time is still being studied from a different perspective, since it is a problem that is still without a robust solution. The current guidance systems are based on Doppler effect using sound, no commercial device uses stereo or RGB-d sensors, because the latter suffer of light saturation in outdoors. In this work, we make use of stereo vision, RGB-d and an IMU to propose a robust guidance system for the visual impaired people. The contribution of this work consists in selecting of state of the art sensors, in considering of cluttered indoors and outdoors scenarios and, in combining creatively simple algorithms to achieve a promising navigation system.

It is of extreme importance for blind and visually impaired people to be independent in their surroundings. This will greatly boost their self-esteem and allow them to have an increasingly comfortable and independent life. The technical aid that we propose is to design a cognitive system based on stereo video cameras and the RGB-D sensor, the use of vanishing points, and fuzzy decision rules, see Fig. 1. In this paper we focus only on the key issue, which is orienting the navigation of the blind in structured environments. Navigation involves an interaction among the systems of stereo video cameras and the RGB-D sensor, the blind person, and the surroundings. Due to static or dynamic reasons, the tasks of guidance are inherently difficult.

Our objective is to orient disabled people to move in structured surroundings with greater efficiency and diminishing the possible dangers that can arise, while eliminating the need for a cane. To do this, we are proposing an approach using stereoscopy to analyse visual space beyond 4 m (recognize objects and signs, detect planes or walls) and using the RGB-D sensor

*Corresponding author: Eduardo Bayro-Corrochano, Department of Electrical Engineering and Computer Science, CINVESTAV Campus Guadalajara Jalisco, México. E-mails: edb@gdl.cinvestav.mx (Eduardo Bayro-Corro-chano), alanda@gdl.cinvestav.mx (Adán Landa-Hernández); hvargas@gdl.cinvestav.mx (Heriberto Casarubias-Vargas).

to get in real time a dense cloud of $3D$ points in a space from 40 cm to 4 m. Using this data, the system calculates the varying vanishing points and applies fuzzy logic for guidance and collision avoidance. The blind individual is oriented following the fuzzy decision rules via an auricular located at the back of the ears. This article is a substantial extension of a preliminary work [1], now including new proposals, like object and sign detection for better management of the navigation planning [13]. In addition, the set of decision fuzzy rules has been tested in a reach variety of scenarios involving lack of points at infinity or signs or the presence of static and moving obstacles.

The paper begins with an introduction and it is followed by second section which describes the perception system: the image acquisition, the detection of lines and points at infinity as well as the detection and stabilization of the floor. The third section is devoted to explain the the use of the RGB-d sensor and the Histograms of Gradients together with a linear support vector machines for the detection of planes, contours of static objects, walking persons and QR code signs. The limits of the orthogonal planes and lower contour of he obstacles are used to draw a free path for walking within a polygon. The fourth section explains the developed set of fuzzy rules for the command the guidance system. The fifth section presents the experimental analysis and the sixth section is devoted to the conclusion.

2. Perception system

2.1. Vision system

The intelligent guidance system consist of: a stereo system (480x640 CCD cameras) together with a RGB-D sensor, an IMU and a pocket computer, see Fig. 2. The 3D data of the stereo system is related with the dense 3D data of the RGB-d sensor, in order to see near with precision and far approximately 15 mts. Since outdoors the RGB-D sensors is saturated due to the strong light, the stereo system is used to support the navigation [12].

The processing outputs guidance commands to the blind ears via audio signals.

After the image acquisition the data (information) processing takes place: the floor is identified, for the navigation planes, obstacles and moving persons are detected. Then the negotiation takes place using a fuzzy inference engine. The long term memory and the cognitive system are in current development, they storage a prior information and ontologies to ensure a robust intelligent guidance systems.

More technical details of the system will be given in next sections.

2.1.1. Image processing

The information processing involves computing the vanishing point for the front. Then it starts to search for planes that are within the range of 1.5-4.0 mts., which is the radius in which the blind will have immediate interaction and is the most important radius because it is where the blind can be endangered by bumping into obstacles. Within a space of this radius, an accuracy of ± 15 cm of depth error is guaranteed; however, beyond this radius, the uncertainty of the depth computed by the RGB-D sensor begins to increase. The system calculates the distance of both planes and objects that are within the range of 4 m. The planes and objects are projected on a planar map representing the floor, so one goes from 3D to 2D in order to generate a navigation map. Since we have a temporary $2D$ map of the scene where the obstacles are well localized, the system can apply fuzzy logic rules of the Mamdani type for decision making in order to suggest to the blind how to avoid obstacles and reach a desired goal.

2.1.2. Image acquisition

The frequency of image acquisition has a great influence on the navigation performance and is the reason why a poor rate of images per second will imply fluctuations in the trajectory. Thus, the system processes 25 frames per second, which guarantees a good performance.

2.1.3. Detection of edges

The Canny filter [9] has been used to implement edge detection. This filter acts on images in gray-scale. The Canny filter uses three parameters that define the behaviour of the algorithm. Two of the parameters are defined heuristically: the inferior and superior limits. The third consists of the window size of the Gaussian convolution. We use a 5×5 window. The filter outputs a binarized image, indicating the edges identified. An algorithm to extract characteristics is applied to the binary image to identify lines along walls or natural paths. The Hough transform is used to detect such lines in the image. Once identified the lines, they

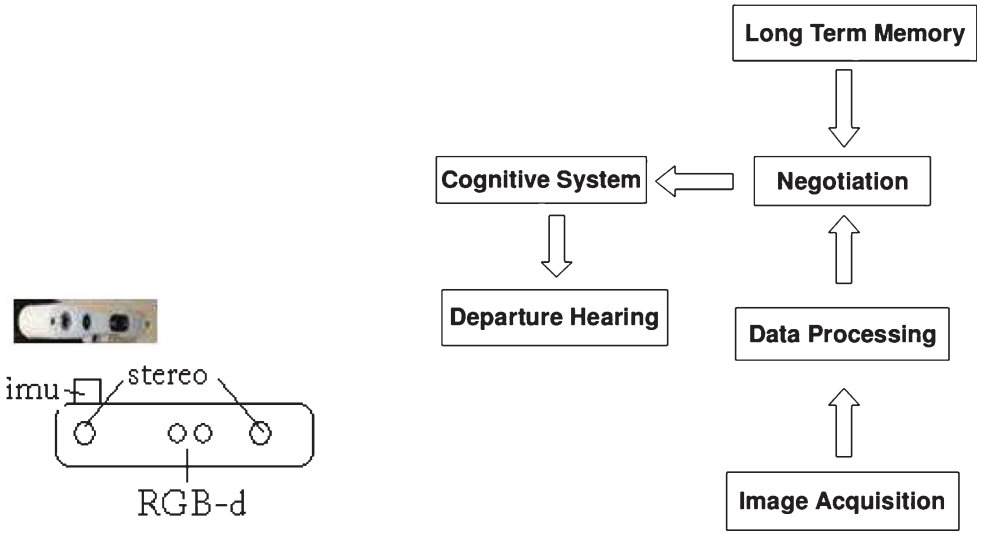

Fig. 1. (left) the guidance system; (right) abstraction of the function of the intelligent guidance system.

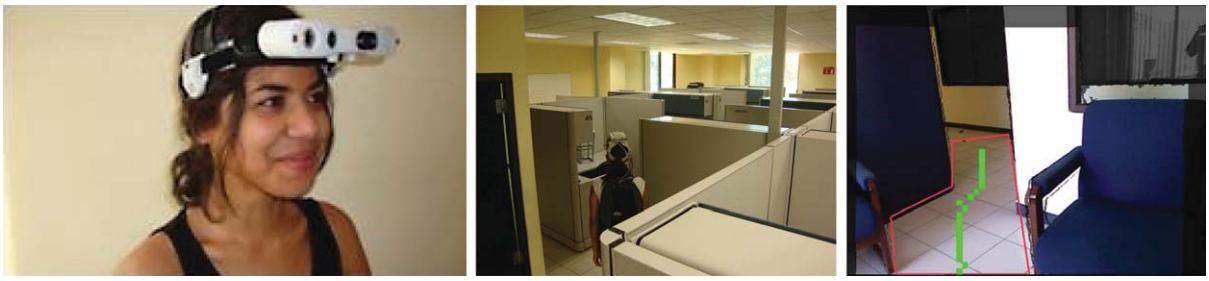

Fig. 2. System for blind guidance: (left) $RBG - D$ sensor with two stereo cameras for $3D$ dense point acquisition at short distances; (middle) navigation; (right) walking path planning.

have to be classified and used to compute the vanishing point.

2.2. The hough transform

This subsection outlines briefly the concept and algorithm of the Hough transform. The Hough transform is the mapping of a line L of $\mathbb{R}^2$ into a single point in the $2D$ Hough space $(\theta,\ \rho)$, where $\rho \in \mathbb{R}$ and $\theta \in [0, 2\pi)$ [10]. The equation describing a line L in $\mathbb{R}^2$ is given by

$$L : y = ax + b, \tag{1}$$

where the scalars a, b are the line parameters. This line L can then be represented in the parameter space as a point with coordinates (a, b). In an opposite way, a point $(x_i, y_i) \in \mathbb{R}^2$ can be represented in the parameter space as a line L' as follows:

$$L' : b = -x_i a + y_i, \tag{2}$$

where $x_i, y_i \in \mathbb{R}$ are the parameters of the line L'. Consider $p_1, p_2, ..., p_n$, a set of points lying on L. Each of these points represents a line in the parameter space. Thus, the intersection of these lines in the parameter space corresponds to the point a_i, b_i, which in turn represents the parameters of the line L in the $\mathbb{R}^2$. For the case of $x = constant$ in Eq. (2), the $y - axis$ coefficient equals 0. One way to go around this problem is to resort to the normal form of the following line equation:

$$L : cos(\theta)x + sin(\theta)y = \rho, \tag{3}$$

where θ and ρ are the parameters of the normal to the line passing through the origin in $\mathbb{R}^2$; see Fig. 3(a). Thus, one point $(x_i, y_i) \in \mathbb{R}^2$ represents a sinusoid in the parameter space, see Fig. 3(b). Thus, the points (x_i, y_i) lying on a line L in $\mathbb{R}^2$ represent sinusoids and

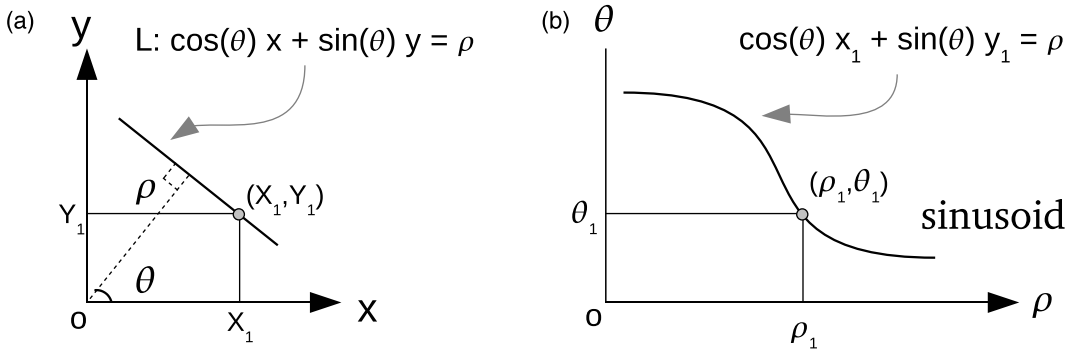

Fig. 3. a) (left) Lines in $\mathbb{R}^2$. b) (right) Lines in $2D$ parameter space.

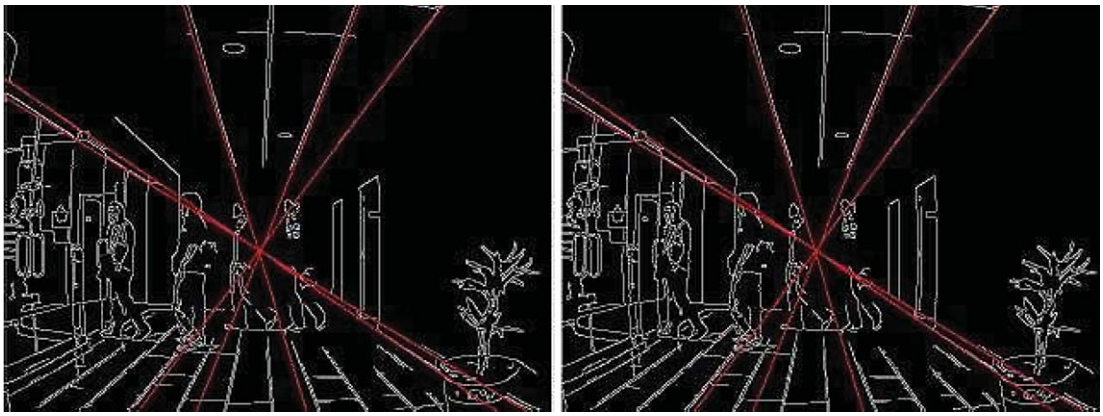

Fig. 4. Detection of vanishing points using lines extracted by the Canny and Hough transform despite the interference of the person's silhouette.

they intersect in just one point $(\theta,\ \rho)$ in the parameter space, see Fig. 3(b). This point represents the line L:$cos(\theta)x + sin(\theta)y = \rho$. So far, we have defined a function that maps a line in $\mathbb{R}^2$ to a point in the Hough space.

2.3. Vanishing point

A vanishing point is a result of the perspective projection of a three-dimensional visual space onto a two-dimensional image plane [11, 14]. Parallel lines in the scene meet at a point at infinity or at the vanishing point. The vanishing point can be used as a virtual compass in 2D images to give a direction to the goal. We regard the primary goal as the place you want to go and a secondary goal as the case when we do not see the main goal; therefore, we guided ourselves with the secondary goal, which would be the vanishing point. If there is neither a visible goal nor a vanishing point, the system looks for a free path within a polygon free of obstacles. For any situation involving guidance, the system uses fuzzy logic for decision making for navigation and collision avoidance.

2.3.1. Calculation of the vanishing point

A point of the projective image plane is represented using homogeneous coordinates by $x = (u, v, 1)^T$, whereas a point of the projective visual space is represented as $X = (U, V, W, Z)^T$. The relationship between x and X is given by projective mapping from the projective visual space to the projective image plane and it is given by

$$\lambda x = PX = K[R|t]X \tag{4}$$

This equation is known as the *pinhole* camera model, where the 3×3 K matrix stands for the intrinsic camera calibration parameters and the 3×4 matrix $[R, t]$ for the extrinsic parameters, which express the $3D$ rigid motion of the camera with respect to a world coordinates system. When the camera geometry is known, each vanishing point corresponds to an orientation vector $[x, y, z]^T$ in the scene, and vice versa.

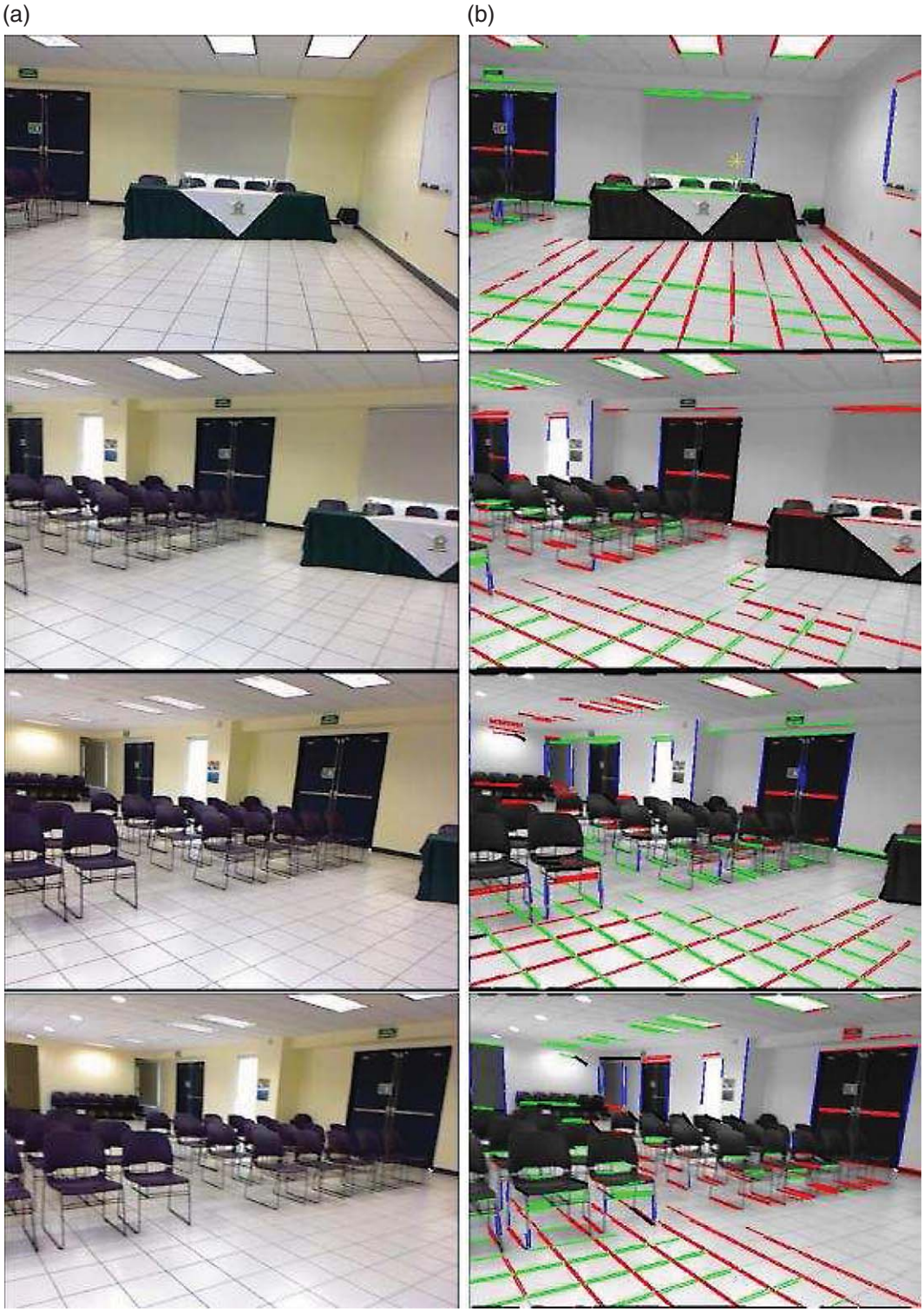

Fig. 5. a) (left column) image secuence; b) (right column) 3D orientation compass: extraction of points at infinity oriented to the front, lateral and upwards. Note the three sets of parallel lines.

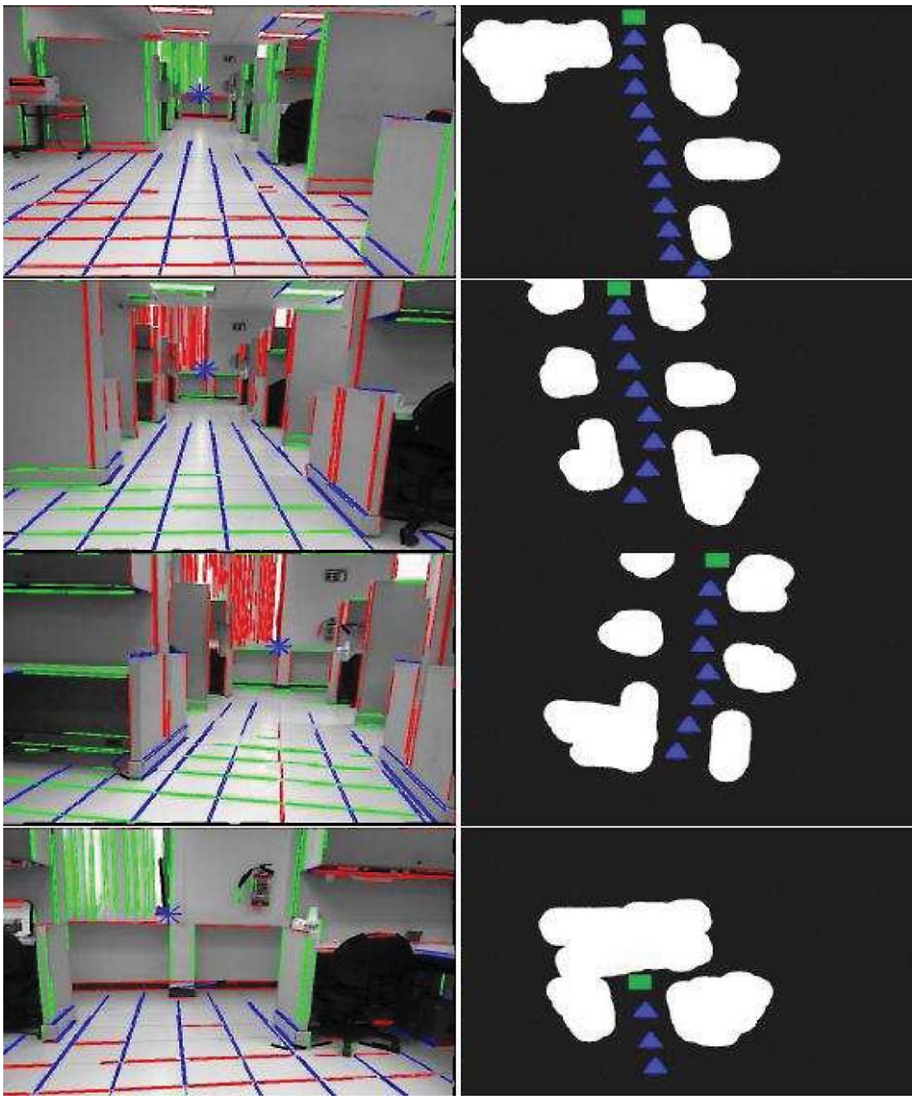

Fig. 6. Sequence of the computation of the point at infinity at front during motion of the guidance system.

If you just know the intrinsic parameters of the camera, one needs to know the external calibration given by the vanishing points corresponding to any two of the axes of the coordinate system of reference; the remaining axis is simply the cross product of the two. One computes the point at infinity in the projective image plane using points of the image that lie on lines. These lines are parallel and correspond to wall intersections, as shown in Fig. 4. The lines are extracted using the Canny edge detector and selected using the Hough transform [4]. The cells in the Hough space with higher voting correspond to large lines. In the projective plane, lines $l_i \in P^2$ are computed using any projective pair of points $x_{i1}, x_{i2} \in P^2$ lying on these lines:

$$l_i = x_{i1} \times x_{i2}; \tag{5}$$

and the intersection of any pair of lines l_i and l_j is given by

$$x_\infty = l_i \times l_j, \tag{6}$$

which in turn represents the point at infinity x_∞. Since an image suffers from the effects of illumination

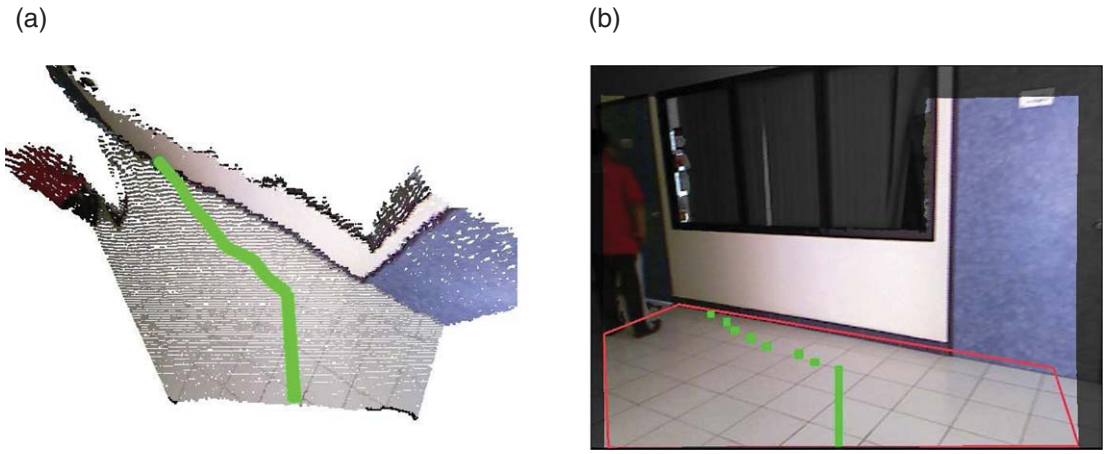

Fig. 7. (a) (left) 3D structure: the floor is identified and the path is computed; (b) (right) path at the polygon of free space avoiding a person.

changes and noise on line borders, we are required to average the intersection of N pairs of parallel lines to compute a robust point at infinity:

$$\hat{x}_\infty = \frac{1}{N} \sum_{i,j=1,N} l_i \times l_j. \tag{7}$$

Note that the Hough transform is used to detect the more important lines, rejecting lines with few supporting points. Taking those parallel lines, we compute an average of their intersection using the equation (7). Those lines along a corridor indicate the correct orientation for the point at infinity at front. In certain cases when one gets lines orthogonal to the corridor e.g. due to floor blocks, these lines help to recover the three points at infinity which serve as orientation compass for the navigation. Figure 5 shows clearly how at different orientations the three orthogonal points at infinity vary. These points at infinity lateral, forward and upwards can be used as orientation compass for guidance towards these orientations. Of course when there is lack of lines the detection of such points at infinity is jeopardised. In our approach, after the vanishing point is computed and since one can detect points on the parallel lines, we resort to the $3D$ information of the RGB-D sensor to get the $3D$ Euclidean orientation, which in turn is projected onto the $2D$ floor image; see Fig. 17. This projected point will constitute the navigation goal based on the point at infinity (see the green point at the top of Figs. 15 and 16).

Figure 4 shows the images obtained by the left and right cameras together with the calculated vanishing points. The vanishing point is used as a virtual compass for the navigation. Recall that you may consider an alternative secondary goal: If the perception system cannot see the primary goal, it uses the vanishing point as a secondary goal. Even though a person protrudes from the scene, the system does not recognize him as such, which does not interfere with the virtual compass. However, if an obstacle in motion must be avoided, the system resorts to the use of fuzzy logic decision rules. The proposed fuzzy inference engine will be explained later. As shown in Fig. 4, the vanishing point computed at the left or right image serves as our secondary orientation when you cannot see the main objective; e.g., the door could be the main objective. In Fig. 6, we show how the vanishing point is successively computed while the device is moved towards the end.

If there is neither a visible goal nor a vanishing point, the system searches for a free path within a polygon free of obstacles. This means that once the floor is detected using either stereo or the RGB-d sensor, i.e. the lower plane of coplanar points, a polygon is drawn avoiding obstacles, then and a line is drawn on the polygon towards the goal. This ensures a free navigation path.

2.3.2. *Stabilization of the floor plane*

For the navigation, it is mandatory to identify the floor plane. After the 3D segmentation of planes, the stereo vision system can label them and identify the plane that lies at the lowest spatial position, see [5]. The planes are detected identifying clusters of coplanar points, from them one select the plane with the normal

pointing parallel to the gravity vector which in turn is obtained using the IMU.

As shown in Fig. 7(a), the localization of the floor in real time is carried out successfully. The detected planes are related frame by frame, matching their common coordinate frames, so that the plane of the floor can be considered stabilized while the person is walking. The other planes, which are off the floor plane, can be used as delimiters of obstacles. In this regard, the algorithm can search for a free polygon for path planning. A path traversing the middle of this polygon can be seen as a suitable path for orienting the visually impaired; see Fig. 7(a). We depict on the image the found path as well; see Fig. 7(b). In addition, this strategy can cope with the problem when the system has no point at infinity as the main orientation and it has to resort to searching for a free path along its way. This simple procedure has proved to be practical and very fast to compute.

3. Detection of planes and contours of static and moving obstacles

The vision system identifies geometric entities such as lines, planes, and also obstacle shapes of objects and persons. The Fig. 8(a) shows the detection of planes orthogonal to the floor using the RGB-d sensor. The intersection of these planes with the floor are used to delimit a walking free polygon as shown in Fig. 8(b).

Earlier we explained the way points at infinity and lines are found. For the case of arbitrary contours of static objects or pedestrians that are potential obstacles, we use an image descriptor based on histograms of oriented gradients (HOG) associated with a support vector machine (SVM) classifier. We used this method as well to detect QR-images which after their detection in the scene are traduced for understanding messages useful for the navigation.

The concept of dense and local histograms of oriented gradients is a method introduced by Dalal et al. [2]. This algorithm counts occurrences of gradient orientation in a local part of the image. The method consists of three steps. i) Gradient computation: Compute gradients using two one-dimensional filters: horizontal $[-101]$ and vertical $[-101]^T$. ii) Histograms of orientations: Split the subimage into different cells that are small squares of a predefined size in pixels. For each cell, its histogram of gradients is computed by accumulating votes into bins for each orientation. iii) Cell and block descriptors for a feature vector: Cell histograms are locally normalized depending upon the values of the neighboring cells' histograms. This normalization is done among a group of cells; such a group is called a block. Among the blocks, a certain overlapping is allowed by shifting the cells, which leads to certain information redundancy. Once the normalization step is carried out, all histograms are concatenated in a single feature vector V. For the normalization factor nf for the vector V, one can resort to one of the following normalization schemes:

$$\begin{aligned} \texttt{none}: nf &= 1 \\ \texttt{L1}-\texttt{norm}: nf &= \frac{V}{||V||_1 + \epsilon}, \\ \texttt{L2}-\texttt{norm}: nf &= \frac{V}{||V||_2^2 + \epsilon^2}, \end{aligned} \tag{8}$$

where ϵ is a small regularization constant. It is used to handle vectors of empty gradients. Figure 9 shows the procedure to detect a the silhouette of a person using the HOG technique.

The feature vectors are then used to recognize obstacles present in the walking path. For that a supervised learning technique is used based on the SVM classifier [3, 15]. One uses a set of training image examples with or without pedestrians described by their HOG, to learn a decision function by means of the SVM. Figure 10 shows an abstraction of the system module for feature extraction and classification.

The SVM classifier is a binary classifier algorithm that searches for an optimal hyperplane in a high-dimensional space. Given a training data set $\{\mathbf{x}_k, y_k\} \in X \times \{-1, 1\}$, where x_k are the training vectors V_k and y_k their class labels, the method maps $\mathbf{x}_k$ to a high-dimensional space via the mapping function Φ, and then a decision function $f(\mathbf{x}) = \mathbf{w} \cdot \Phi(\mathbf{x}) + b$ is used to separate the two classes linearly. This function is optimal, because it maximizes the nearest point $[\Phi(\mathbf{x})|1]^T$ and the hyperplane $[\mathbf{w}|b]^T$, here $[\mathbf{w}$ represents the plane orientation. The sign of $f(\mathbf{x})$ determines the class of $\mathbf{x}_k$. This optimization problem can be formulated as a L_1 soft-margin SVM classifier problem as follows:

$$\min_{\mathbf{w},\xi} = \frac{1}{2}||\mathbf{w}||^2 + C\sum_{k=1}^{m} \xi_k, \tag{9}$$

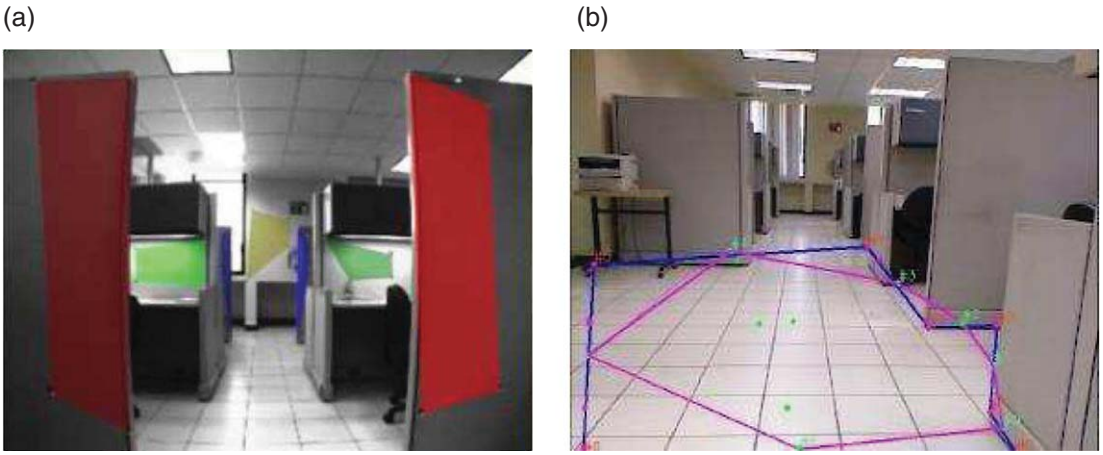

Fig. 8. (a) (left) Planes detection using the RGB-d sensor; (b) (right) polygon of free space avoiding obstacles delimited by the planes.

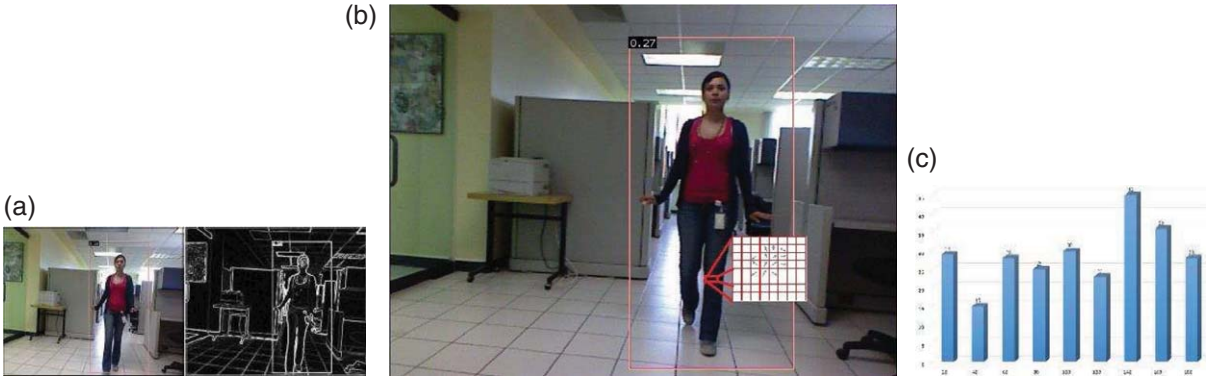

Fig. 9. HOG detection of a person: a) (left) image gradient in the zone of analyse; b) (middle) extraction of gradients; c) (right) computed histograms.

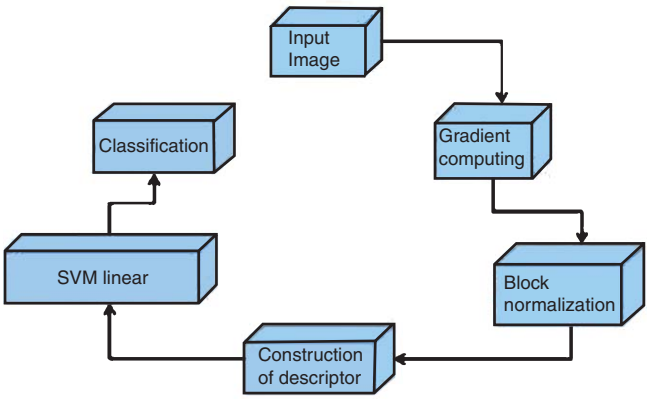

Fig. 10. Feature extraction and classification using HOG and a SVM.

Fig. 11. Detection of arbitrary contours based on HOG and a SVM classifier. (a) (left) A person and a plant are detected; b) a QR-image (c) (right) a QR-announce.

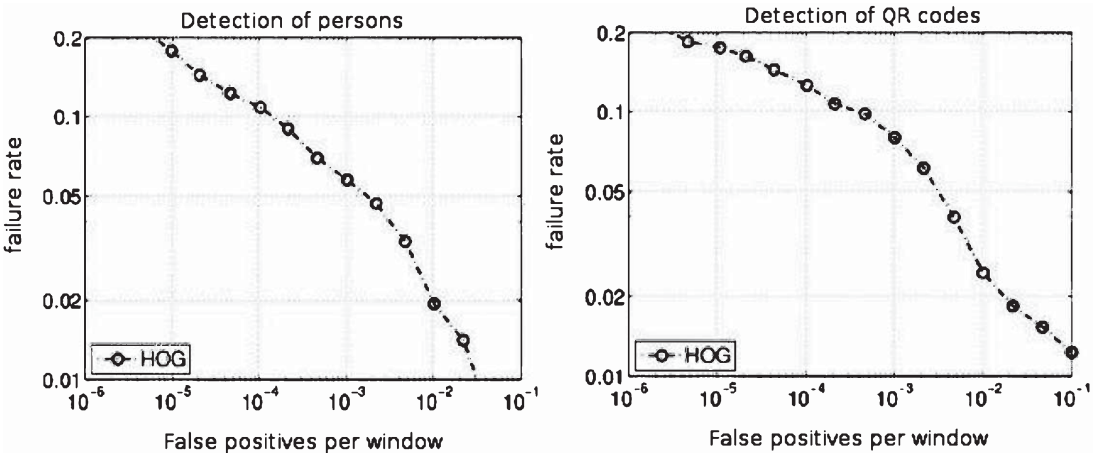

Fig. 12. Performance of the SVM detection of persons and the QR-images.

Fig. 13. Detection of a sequence of arbitrary contours based on HOG and a SVM classifier. (a) (left) A door handle and a plant are detected; (b)(middle) plant, person approaching, and one leaving; (c) (right) a group of students walking.

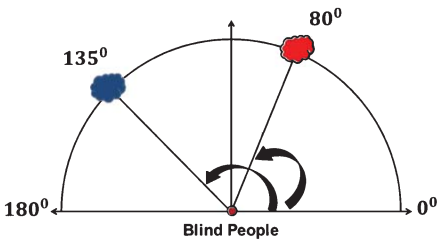

Fig. 14. Sector of orientations.

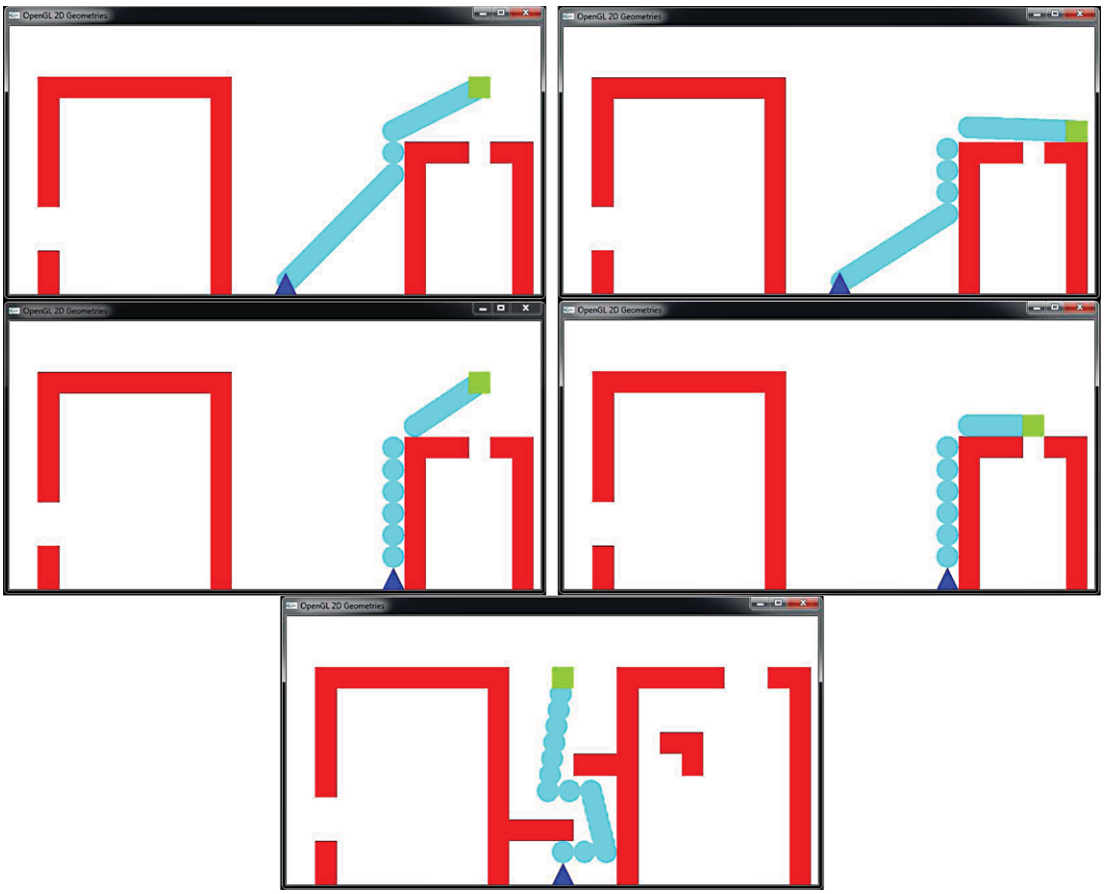

Fig. 15. Examples of path following by different navigation paths.

under the constraint $\forall k,\ y_k f(\mathbf{x}) \geq 1 - \xi_k$. To solve this problem, one resorts to the Lagrangian theory, so that it is possible to compute the weight vector:

$$\mathbf{w} = \sum_{k=1}^{m} \alpha_k^* y_k \Phi(\mathbf{x}_k), \tag{10}$$

where α_k^* is the solution to a quadratic optimization problem given by

$$\max_{\alpha} W(\alpha) = \sum_{k=1}^{m} \alpha_k - \frac{1}{2} \sum_{k,l}^{m} \alpha_k \alpha_l y_k y_l K(\mathbf{x}_k, \mathbf{x}_l), \tag{11}$$

subject to $\sum_{k=1}^{m} y_k \alpha_k = 0$ and $\forall k$, $0 \leq \alpha_k \leq C$, where $K(\mathbf{x}_k, \mathbf{x}_l) = (\Phi(\mathbf{x}_k, \Phi(\mathbf{x}_l)))$. According to Eqs. (9–11), the solution of the SVM problem only depends upon the so-called Gramm matrix $K(\mathbf{x}_k, \mathbf{x}_l)$.

For the classification of objects, we use the following parameters: size of block 2x2; number of bins: 8; size

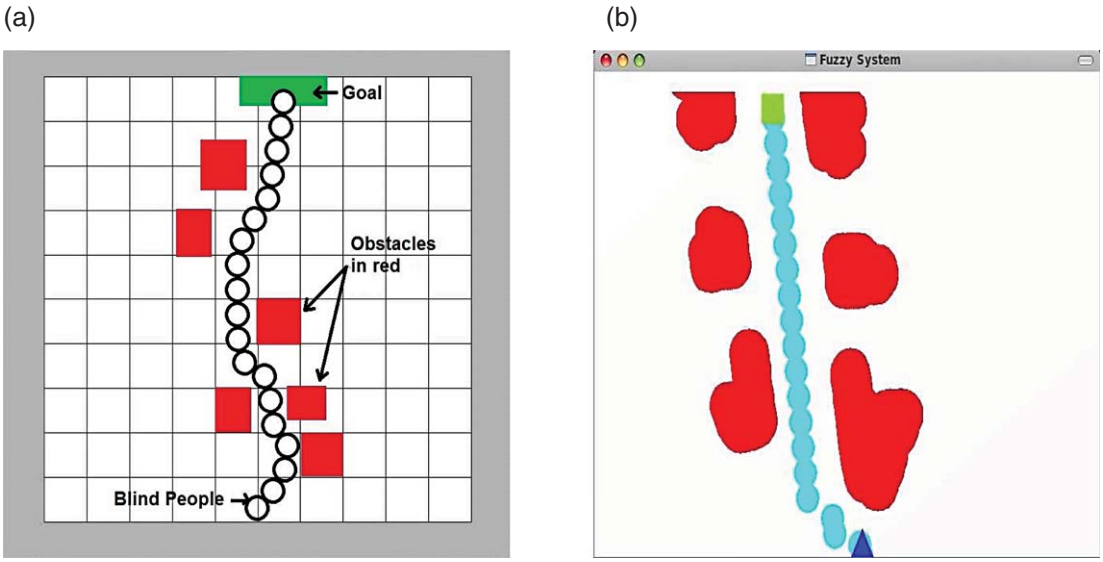

Fig. 16. (a) (left) Example of a navigation path avoiding obstacles; (b) (right) one path generated by the inference engine processing in real time using the stereo data, vanishing points, and the obstacle's location on the floor map.

of cell: 8; overlap of blocks: 1; normalized factor for block: 1.2. We used a SVM C++ code of public domain. The SVM were trained with 100 samples until a very low classification error. The selection of samples was done manually choosing representative images including all sort of obstacles with high gradient. In Fig. 11, we present results of the detection of a walking person and a static object, and a QR image. The SVM performance for the detection of persons and QR code images is presented in the graphics of Fig. 12 where the failure rate decays when the false positives number augments when the size of the window increases. The curves show a good performance for both the person and the QR code detection.

The execution time of HOG and SVM for recall is for real time within the expected walking time of the blind. Since the detection based on HOG consider patches of high gradient changes, illumination changes rather increase the edges in the target and not in the homogeneous areas where there is no targets, as a result the method is robust under illumination perturbations. In Fig. 13, we show the detection procedure working in real time, we show here just three images of the sequence: on the left, we see that the handle of the door and a plant are detected; in the middle, we see an approaching person and one leaving person; and on the right is a group of students walking. The procedure can detect several targets simultaneously in real time fast enough to guide the blind.

4. Fuzzy inference system

This section gives a brief outline of fuzzy logic and explains a procedure for fuzzy decision making using a Mandami fuzzy decision scheme [6–8].

4.1. Fuzzy logic

Fuzzy logic is used when the complexity of the process in question is very high and there are no accurate mathematical models for nonlinear processes involving wrapped definitions and imprecise or subjective knowledge. These fuzzy systems are generally robust and tolerant to imprecision and noise in the input data, thus showing the output's speed and accuracy.

4.2. Mandami fuzzy decision system

Figure 14 shows the configuration of the angles to the objects and which overhead obstacles can cause a hazard to the visually impaired. In our proposal, the obstacles having a high collision risk are represented with small angles, meaning that one has to take extra care to avoid an accident, as opposite larger angles for those objects are considered lower risk. Next, we list a set of the fuzzy rules we are proposing to help a blind person navigate in a structured indoor environment.

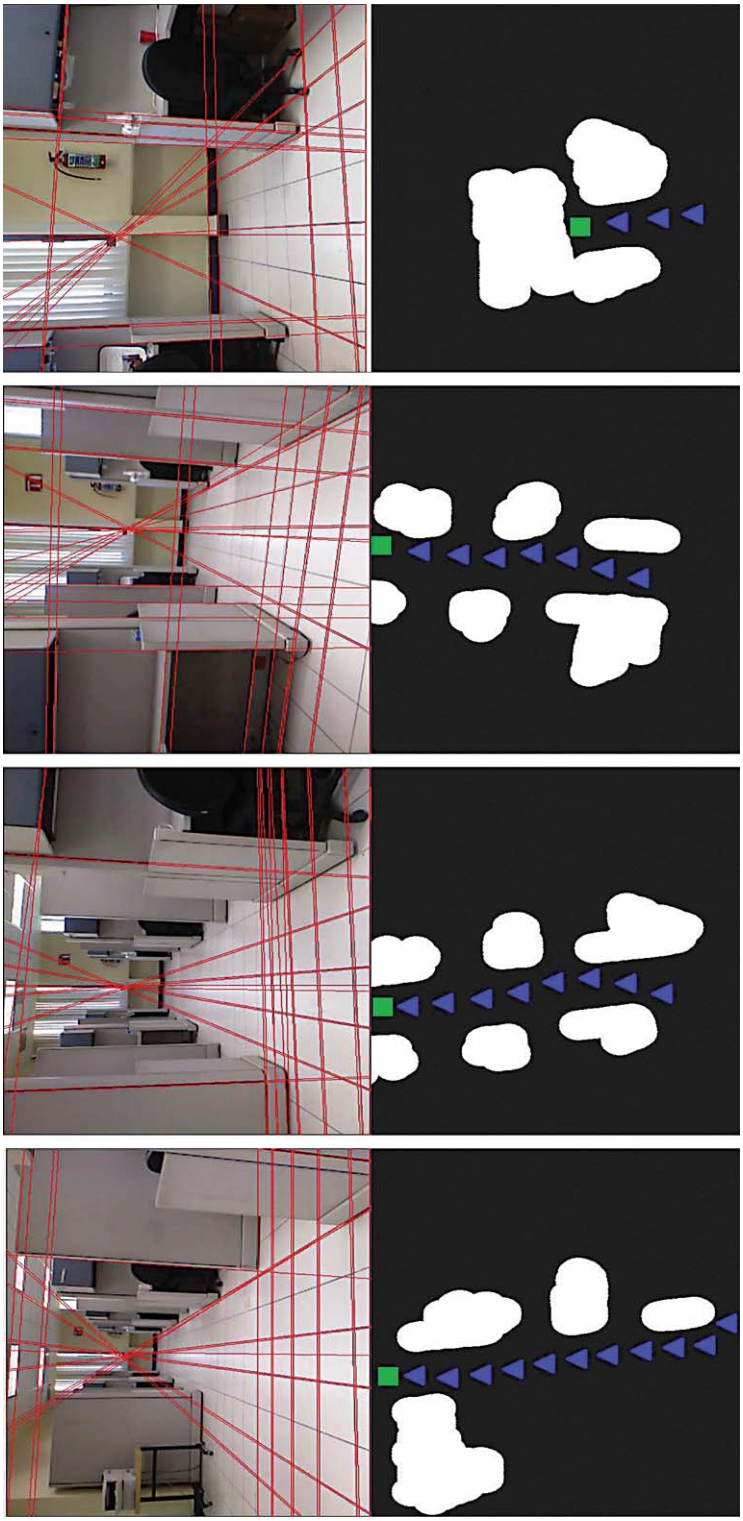

Fig. 17. Using the point at infinity: views at different instances of the path for displacement of the blind individual. On the upper row: the lines extracted by the Canny detector and Hough tranform and the computed points at infinity; on the lower row: the $2D$ maps of the obstacles with respect to the floor and the paths computed with the visual hints and the fuzzy inference engine.

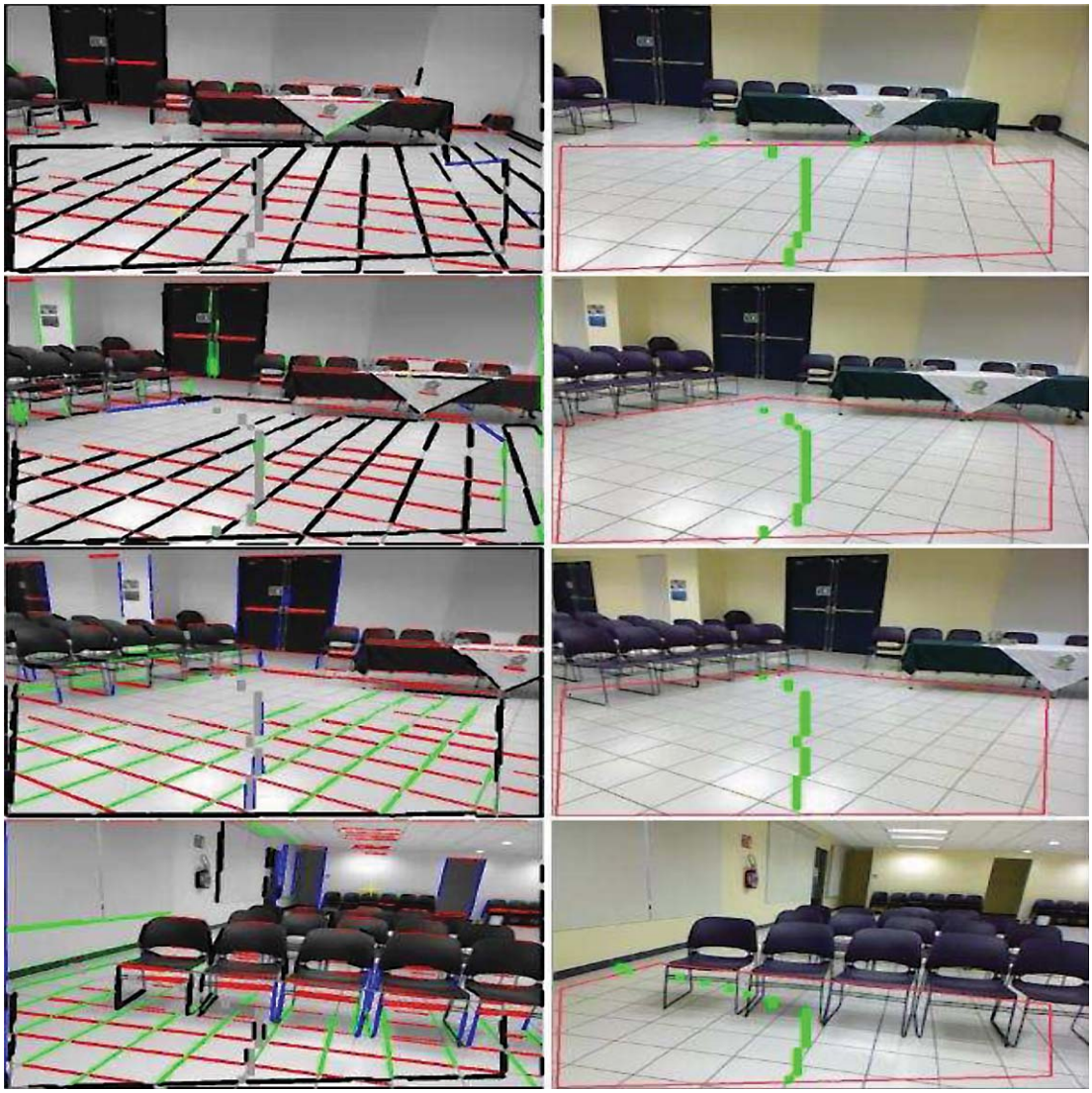

Fig. 18. Navigation path avoiding obstacles: a succession of paths generated by the inference engine processing in real time using the stereo data, vanishing points, and the obstacle's location on the floor map.

Acting as knowledge engineer, this set was obtained intuitively by hand proposing more like natural rules and after tests pruning the redundant ones, so that we achieved a set of reduced number of rules. In the experimental part we show various tests of this set in variety of scenarios with orienting points at infinity or without them also with static obstacles or walking persons. The set of the fussy rules is as follows:

1) If (Length is Near) and (Angle is Tiny) and (Speed is Middle) and (VanishingPoint is Center), then (Left is Medium)(Forward is Slow).
2) If (Length is Near) and (Angle is Center) and (Speed is Slow) and (VanishingPoint is Center), then (Left is Medium)(Forward is Slow).
3) If (Length is Near) and (Angle is VeryBig) and (Speed is Middle) and (VanishingPoint is Center), then (Forward is Slow)(Right is Medium).
4) If (Length is Middle) and (Angle is Tiny) and (Speed is Middle) and (VanishingPoint is Center), then (Left is Medium)(Forward is Slow).
5) If (Length is Middle) and (Angle is Small) and (Speed is Slow) and (VanishingPoint is Center), then (Left is Slow)(Forward is Slow).

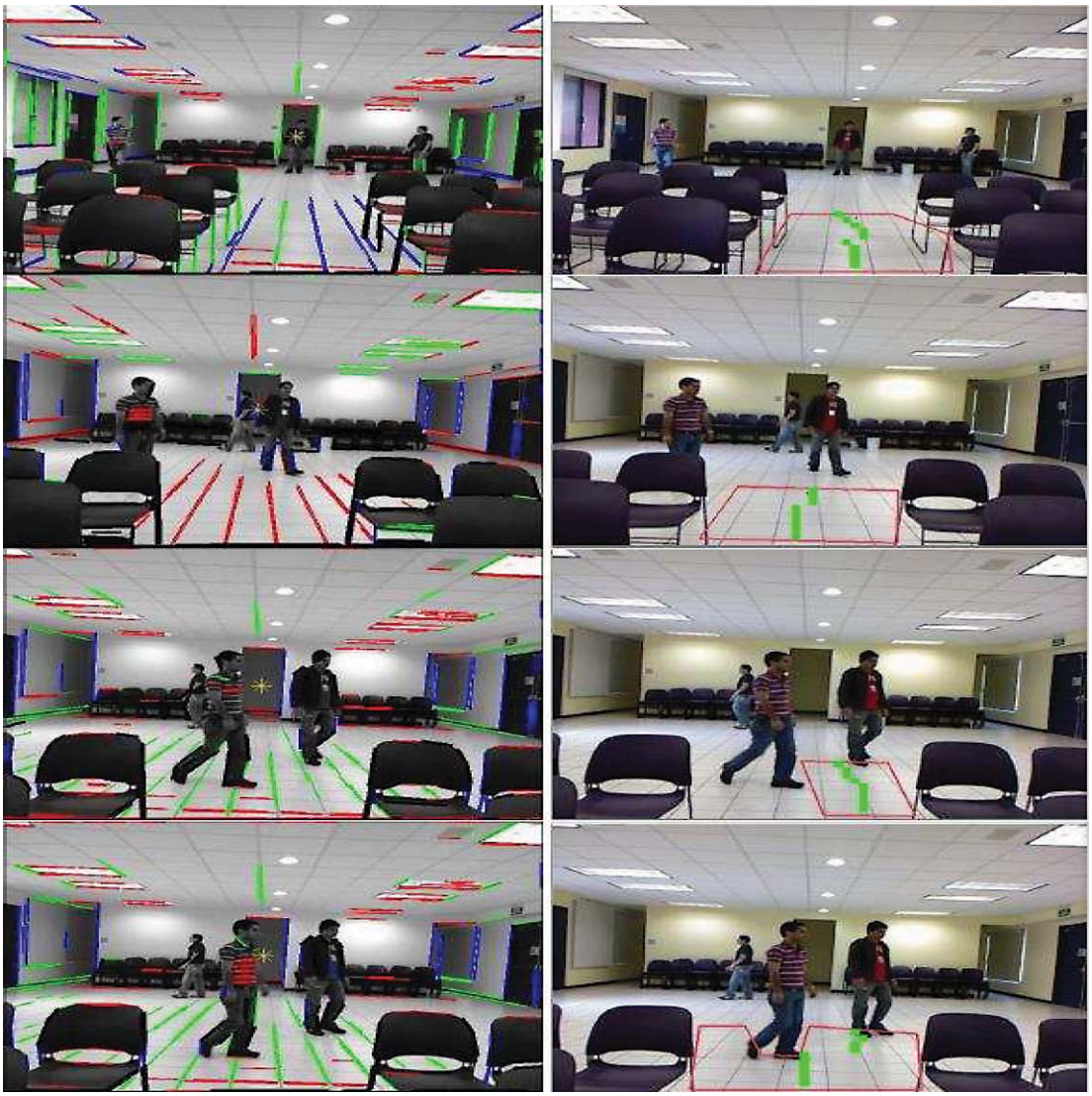

Fig. 19. Navigation path avoiding obstacles and walking persons: a succession of paths generated by the inference engine processing in real time using the stereo data, vanishing points, and the person and obstacle's location on the floor map.

6) If (Length is Middle) and (Angle is Big) and (Speed is Middle) and (VanishingPoint is Center), then (Forward is Slow)(Right is Fast).
7) If (Length is Far) and (Angle is Big) and (Speed is Slow) and (VanishingPoint is Center), then (Forward is Medium)(Right is Slow).

With the use of these proposed rules, Fig. 14 depicts where the object is: at an angle of 80° with respect to the horizontal at close range. Then the fuzzy decision system uses rule 2, in which just two options are left to navigate: One of them is to move to the left with a moderate speed because this way has a danger of colliding; the second option is to move forward slowly, as there is a little more impact and risk close to the blind. However, whether to take a step forward is re-evaluated at the scene, and if at that time the object is already at a safe distance from the blind person, then the system indicates to the blind that he or she can only move left quickly, and so on, until he or she reaches the desired goal. The distance is between 0-300 cm. in diameter; however, we only consider the half-circle in front of

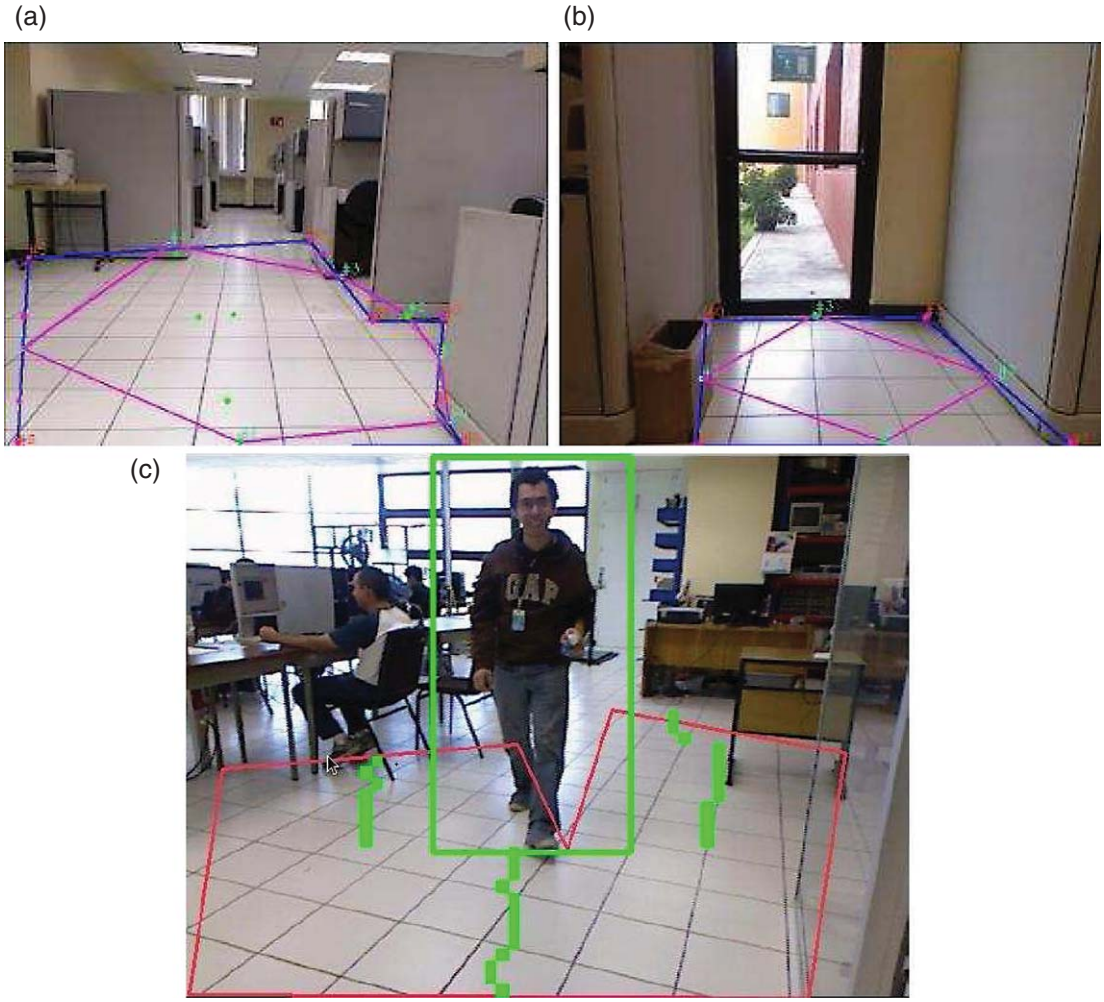

Fig. 20. Finding a path free using a polygon: a) (upper left) in an office; b) (upper right) in front of a door; c) (lower row) using SVM-HOG in front of a walking person.

the blind or visually impaired, which is his or her area of interaction. As we can see from the image, three areas of interest are being considered: These are the case when the objects are close to half of the greatest distance to be covered. In the series of images shown in Fig. 15, we show results of a simulation of representative navigation paths generated by the fuzzy inference engine. In Fig. 16(a), we show a navigation trajectory in a simulated environment similar to an office.

5. Experimental analysis

In order to generate the path, the inference engine identifies hazard areas; see the red colour in Fig. 16(b). Each area is expanded to create a secure surrounding so that the blind can then pass by without hitting any obstacle. Whether this area is a person or an obstacle, the inference machine has to choose among many possible paths the one of the shortest distance between the target and the blind person. In general, the system selects just one path. In this respect, the fuzzy rules are formulated in such a way by an experienced knowledge engineer that the computed navigation decision is most appropriate for the blind.

5.1. Guidance with a point at infinity

In Fig. 17, we can appreciate the real-time processing of the images; the upper row shows the extraction

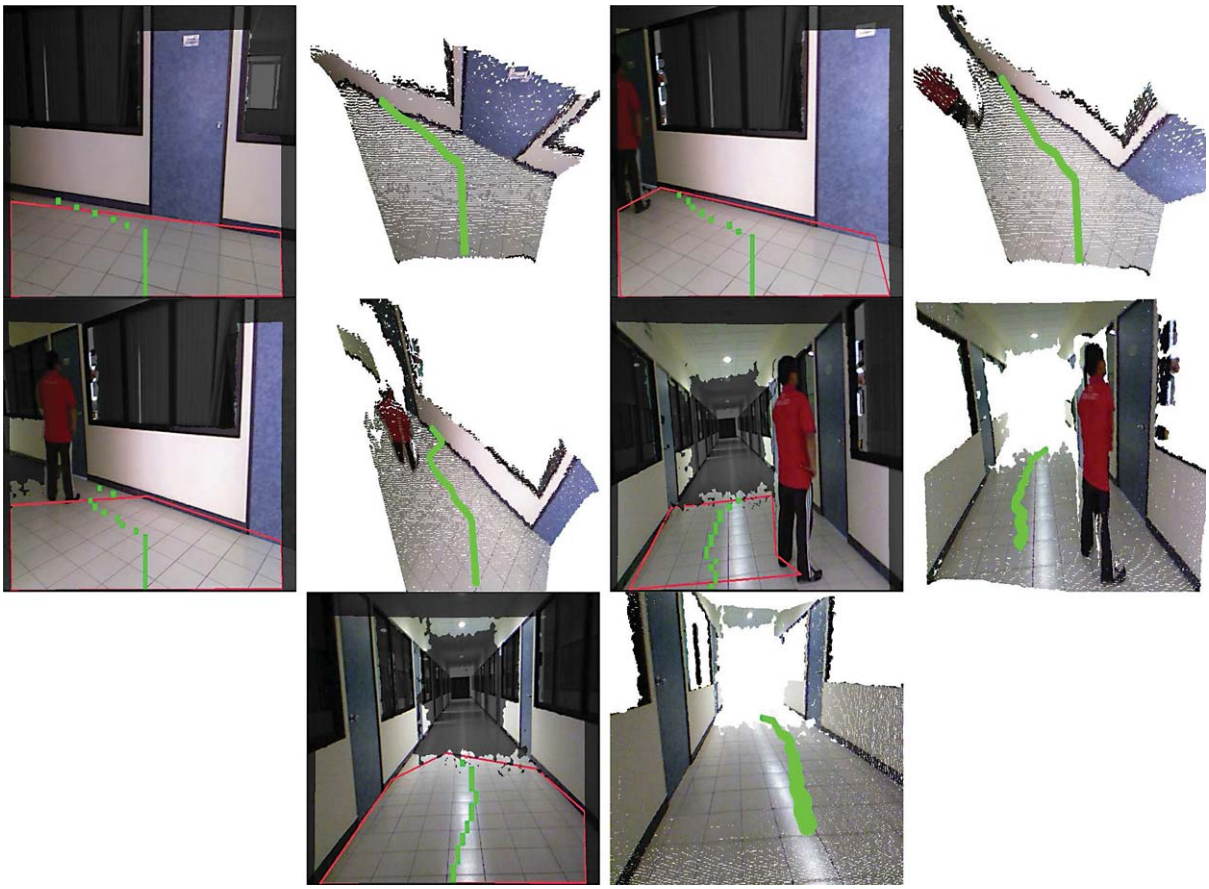

Fig. 21. Sequence of navigation towards an office: planning paths on the floor detected. The path is drawn on the images and also on the 3D virtual-world representation.

of the vanishing point for the guidance. Both the current vanishing point and the binary image information of the floor are fed into the inference engine, which in turn computes the free walking path. As can be seen on the lower row, the inference engine indicates to the blind a free path to avoid obstacles. The first image from the right in the upper row shows that due to the lower number of parallel lines, the computation of the point at infinity is less accurate, which affects the path planning by the inference engine.

We have also tested the system in a place where there is no sign or QR-image to orient the guidance device like in a big closed room. You can see in Fig. 18, the system using vanishing points finds its paths for navigate. In Fig. 19, how the system avoids walking persons for continuing moving towards the end of the class room.

5.2. Guidance using a free path on a polygon

When it is not possible to compute the point at infinity, we resort to following the middle of a polygon representing a free space for walking, as presented in Fig. 20(a,b). In Fig. 20(c), the procedure first detect via SVM-HOG the walking person, then the deliming planes to draw a path free polygon. Figure 21 shows the case when the blind is going into an office; note that a person was detected and circumvented. In Fig. 22, the blind goes into the office, two ways of approaching the table are indicated, and the last images show when the person leaves the office. In both sequences, we can appreciate the effectiveness of our approach for guiding the blind to avoid collisions with objects or persons and using either the orientation of the point at infinity or the polygon of free space. The fuzzy engine is

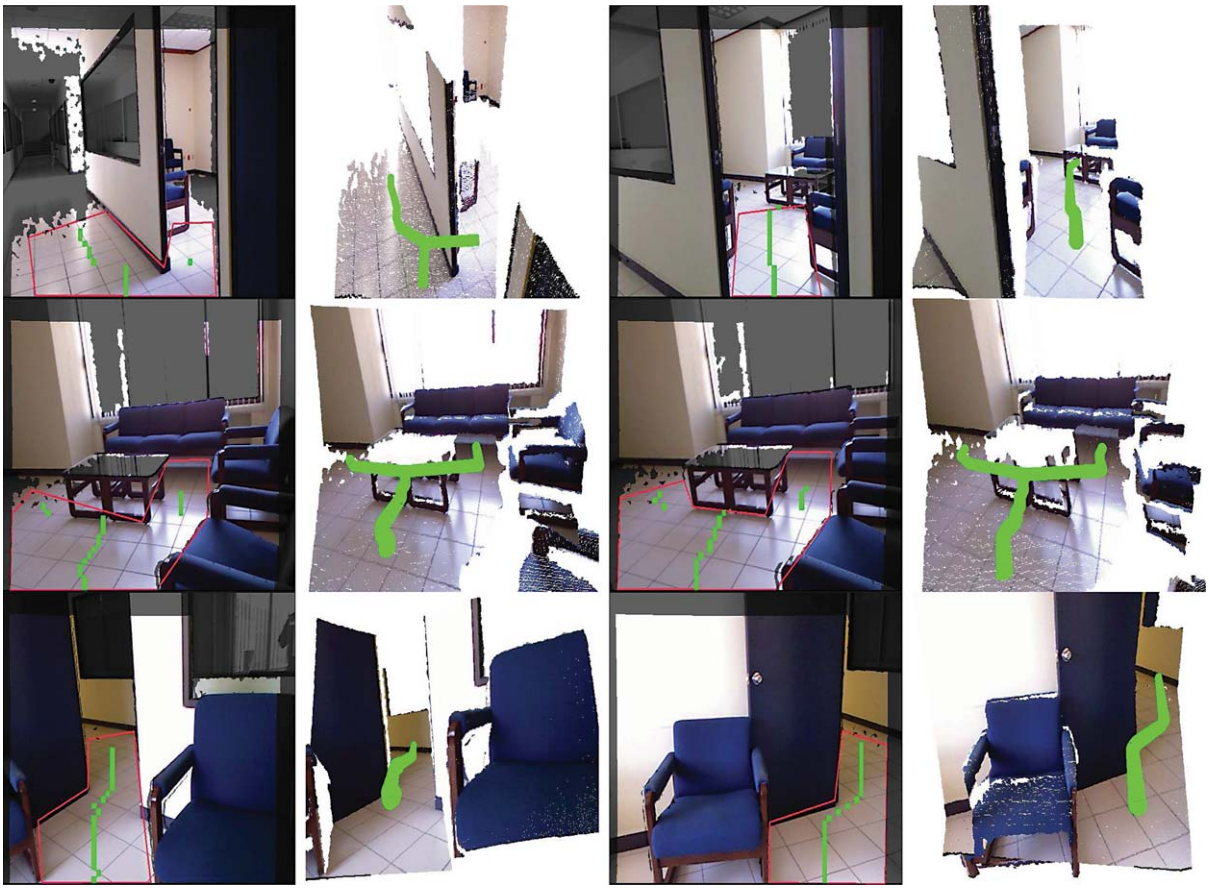

Fig. 22. Sequence of entering and leaving an office: planning paths with detected floor. The path is drawn on the images and on the 3D reconstruction.

used when the path is in a big area obstructed by many obstacles. The approach based on the polygon is used for small spots and used by the inference engine as well to navigate within such small parts of big areas.

5.3. Performance analysis

Since the use of the vanishing point depends on the existing lines of walls or a large corridor, it is not enough in free spaces or when the blind individual has to turn around a corner. For that we will enlarge the perception capacities of the system to detect spatial landmarks such as signs, RQ-codes, or traffic lamps. We are also improving the stabilization of the reconstructed walking plane and its registration through the frames. To do so, we are computing a plane with the RANSAC algorithm and implementing a Kalman-filter based SLAM that uses gyroscope measurements so that the coordinate frame of the blind with respect to the plane will be well estimated. This also offers the possibility to grow a temporal map (short-term memory), which can be reused from time to time by the blind, for example, to come back through an area already visited.

6. Conclusions and future work

The contribution of this work consists in selecting of state of the art sensors, in considering of cluttered indoors and outdoors scenarios and, in combining creatively simple algorithms to achieve a promising navigation system. We conclude that the method shows that under certain reasonable conditions, the stereo vision with the vanishing point fusion and the Mamdani-type fuzzy logic system is enough to help the guidance of the blind. This work has presented also a facility of the device to detect: planes, contours like walking persons or non rigid obstacles as well as the

detection of QR code images. Once the planes orthogonal to the floor are detected, they are used together with the contours of both static and walking persons to build a free navigation polygon which in turn is used to guide the blind. The guidance can follow the orientation of points at infinity or a sign like of a QR code or in absence of those simply follow the free path within the polygon. Due to the faster processing time, the system manages in real time to determine the path that will lead to the desired goal. However, there are still obstacle configurations that can cause problems since we cannot cover all possible configurations. So far only a reduced number of fuzzy rules have been considered; they have been however worked out in 80 different configurations. The time demanded due to evaluating all the rules and finding the nearest trajectory is however in the millisecond range.

We continue working with this project to complement it with some other technique to make it more robust and cover more configurations to further secure the navigation of the blind. In order to build a much more robust guidance system, we will increase the perception capacities of the system to detect spatial landmarks such as signs, or traffic lamps. Since the RGB-d sensors are saturated in outdoors, we will improve the stereo vision module for a better detection of walking paths under over illumination like in outdoors. Furthermore, we will improve the stabilization of the reconstructed walking plane and its registration through the frames. In order to do so, we are computing a plane with RANSAC and we will use a Kalman filter for SLAM that uses gyroscope measurements so that the coordinates frame of blind with respect to the floor plane will be better estimated. In addition, in order to cope the lack of points at infinity or signs around, the guidance device will resort to a digital map to assist in the orientation of the path free within the polygon. These new results will be presented in the near future.

References

[1] A. Landa-Hernandez and E. Bayro-Corrochano, Cognitive guidance system for the blind, In *Proc. of the World Automation Congress*, WAC'2012, Pto. Vallarta, Mexico, August 2012, pp. 1-6.

[2] N. Dalal and B. Triggs, Histograms of oriented gradients for human detection, In *Proc Int Conf Computer Vision Pattern Recognition*, **2**, Rhone-Alpes, 2005, pp. 886-893.

[3] B.E. Boser, L.M. Guyon, and V.N. Vapnik, A training algorithm for optimal margin classifiers, In D. Haussler, editor, *5th Annual ACM Workshop on COLT*, Pittsburgh, PA, ACM Press, 1992, pp. 144-152.

[4] C.A. Kessler, C. Frietsch, N. Weinmann, M. Trommer and F. Gert, Vision-based attitude estimation for indoor navigation using vanishing points and lines, *Position Location and Navigation Symposium (PLANS)* (2010) IEEE/ION, pp. 310-318.

[5] C.-H. Lin, S.-Y. Jiang, Y.-J. Pu, and K.-T. Song, Robust ground plane detection for obstacle avoidance of mobile robots using a monocular camera, *The 2010 IEEE/RSJ International Conference on Intelligent Robots and Systems*, October 18–22, 2010, Taipei, Taiwan.

[6] M. Wang and J.N.K. Liu, Fuzzy logic-based real-time robot navigation in unknown environment with dead ends, *Robotics and Autonomous Systems* **56** (2008), 625-643.

[7] E.H. Mamdani, Application of fuzzy algorithms for the control of a simple dynamic plant, *Proc. IEEE*, (1974), 121-159.

[8] E.H. Mamdani and S. Assilian, An experiment in linguisic synthesis with a fuzzy logic controller, *International Journal of Man – Machine Studies* **7**(1) (1975), 1-15.

[9] J.A. Canny, A computational approach to edge detection, *IEEE Trans. Pattern Analysis and Machine Intelligence* **8**(6) (1986), 679-698.

[10] R.O. Duda and P.E. Hart, Use of the Hough transformation to detect lines and curves in pictures, *Communications of the Association for Computing Machinery* **1972** (15), 11-15.

[11] Z.-W. Kim, Geometry of vanishing points and its application to external calibration and real-time pose estimation, *Research Report*, 07-01-2006.

[12] S. Nogueira, Y. Ruichek and F. Charpillet, *A self-navigation technique using stereovision analysis*, In A. Bhatti, editor, Stereo Vision, In Tech, 2008.

[13] G.N. DeSouza and A.C. Kak, Vision for mobile robot navigation: A survey, *IEEE Transactions on Pattern Analysis and Machine Intelligence* **24**(2), February 2002.

[14] S.T. Barnard, Interpreting perspective images, *Artificial Intelligence* **21**(4) (1983), 435-462.

[15] V. Vapnik, *Statistical Learning Theory*, Wiley, New York, 1998.