Univ Access Inf Soc (2015) 14:67–80
DOI 10.1007/s10209-013-0338-8

LONG PAPER

Computer vision and GIS for the navigation of blind persons in buildings

M. Serrão · S. Shahrabadi · M. Moreno · J. T. José · J. I. Rodrigues · J. M. F. Rodrigues · J. M. H. du Buf

Published online: 19 February 2014

Abstract This paper presents a system which integrates a geographic information system of a building with computer vision. It uses only one camera, for example, the one of a mobile phone. Visual landmarks, such as frontal and lateral doors, stairs, signs, and fire extinguishers, are employed for localizing the user in the building and for tracing and validating a route for the user's navigation. The developed system clearly improves the autonomy of persons with a very low vision during indoor navigation.

Keywords Navigation · Localization · Accessibility · Autonomy · Blind · Vision · Geographic information system

1 Introduction

Worldwide, approximately 285 million persons are visually impaired. About 39 million are completely blind and 246 million have low vision [30]. There are many types of visual impairments and official definitions in terms of reduced acuity and field of view. Below, the word blind is used to refer to persons who may have some residual vision but who must rely on—or simply feel more comfortable when using—the white cane, or, if they can afford one, a guide dog. The white cane serves *local* navigation, by constantly swaying it in front for negotiating walking paths and obstacles in the immediate vicinity. In this paper, *global* navigation is addressed, going to a specific location in a large building. If there are no Braille signs and the building has not yet become familiar, blind persons, even when accompanied by a guide dog, must rely on people passing by to ask for information. Apart from guiding a user to a destination, a global navigation aid can provide the user with important landmarks for creating a more complete mental map of the building, therefore improving autonomy in the future.

Different assistive technologies exist or have been proposed. One named smart cane [26] is an electronic cane with built-in ultrasound sensors for detecting obstacles, both horizontally and vertically. It is planned to be cheap such that it can be afforded by poor persons. Drishti [20] is an in- and outdoor navigation system. Outdoor it uses DGPS localization to keep the user as close as possible to the central line of sidewalks. It provides the user with an optimal route by means of its dynamic routing facility. The user can switch the system from out- to indoor operation with a simple vocal command, which activates a precise ultrasound positioning system. In both cases, the user receives vocal prompts which alert for possible obstacles and provide guidance while walking about. CASBliP or Cognitive Aid System for Blind People [3] was a European project. Its main goal was to develop a system capable of interpreting and managing real-world information from different sources in order to improve autonomous mobility. Environmental information from various sensors is acquired and transformed into enhanced images for

M. Serrão
Instituto Superior de Engenharia, University of the Algarve, 8005-139 Faro, Portugal
e-mail: migps.g@gmail.com

S. Shahrabadi · M. Moreno · J. T. José · J. M. F. Rodrigues (✉) · J. M. H. du Buf
Vision Laboratory, LARSyS, University of the Algarve, 8005-139 Faro, Portugal
e-mail: jrodrig@ualg.pt

J. M. H. du Buf
e-mail: dubuf@ualg.pt

J. I. Rodrigues
CIMA, University of the Algarve, 8005-139 Faro, Portugal

visually impaired users, or into acoustic maps via headphones for blind users. SmartVision: active vision for the blind [6] was a Portuguese project. Its prototype is an in- and outdoor navigation system with different modules which integrate GPS and WiFi localization with a geographic information system (GIS) database, passive RFID tags in sidewalks, and computer vision for path centering and obstacle avoidance.

Schmitz et al. [23] developed a navigation system that seamlessly integrates static maps with dynamic location-based textual information from a variety of sources. Each information source requires a different kind of acquisition technique. All acquired information is combined by a context management platform and presented to the user as a tactile or acoustic map depending on the sources available at a current position. Positioning is achieved by a combination of an inertial tracking system, RFID technology, and GPS, and the user is guided to a desired destination by speech output and a haptic cane. Costa et al. [5] presented an algorithm to recognize landmarks suitably placed on sidewalks. The proposed algorithm uses a combination of Peano–Hilbert space-filling curves for dimension reduction of image data and ensemble empirical mode decomposition (EEMD) to preprocess images, which yields a fast and efficient recognition method.

This paper presents a system which integrates data in a GIS of a building with detection of visual landmarks. Any normal camera can be used together with a small, portable computer such as a netbook. GIS/vision-based localization is complemented by navigation: At any time, the system traces and validates a route from the current position to a given destination. Although designed for being integrated in the Blavigator prototype, see Sect. 2, this system can be used by any person who wishes to navigate in a complex building. The main contributions are the integration of existing GIS data with visual landmarks (staircases, doors, fire extinguishers, etc.) for localization and navigation, and the detection of stairs and lateral doors in corridors. For previous work on the detection of doors, the reader can refer to [4, 18, 19, 28], for stairs to [12, 13, 15, 16, 24, 31], and for object recognition to [21].

The rest of this paper is organized as follows. The following section describes the prototype and the geographic information system. In Sect. 3, landmark detection, including doors and stairs, is explained. Section 4 deals with navigation, i.e., route planning and localization during navigation. Final conclusions are presented in Sect. 5.

2 The prototype and the geographic information system

The system presented here is part of a larger project entitled "Blavigator: a cheap and reliable navigation aid for the blind". The goal is to develop a vision and navigation aid which is (a) not expensive, given that about 90 % of potential users live in so-called developing countries; (b) easily portable, not being a hindrance when walking with the cane; (c) complementing the cane, but not substituting it because blind persons must always be able to rely on the cane; (d) extremely easy to use in terms of intuitive interfacing; (e) simple to assemble, install, and operate, without need for very skilled technicians; and (f) providing useful assistance for local and global navigation in real time. Blavigator, a synthesis of the words "blind" and "navigator", is a follow-up project of the above mentioned SmartVision [6].

Geographic information usually links locations to properties of those locations. Technologies for handling such information include GPS, remote sensing, and geographic information systems (GIS) [10]. In the prototype presented, any detected landmarks (doors, stairs, fire extinguishers, exit, and WC signs, etc.) are matched against those in the GIS for each room, corridor, etc. By combining the detected landmark positions with the traced route, the user can be informed about the current location by a speech module. Since the GIS/landmark system must be integrated in the SmartVision/Blavigator prototype, all visual functions have been optimized for small CPU time and memory usage. Most visual functions are based on a few basic algorithms, which are employed only once for each video frame.

For the validation of the concept, the prototype was developed on a netbook with a standard WiFi webcam. It was tested at the Institute of Engineering (ISE) of the University of the Algarve (ISE/UAlg). A mountable support holds the camera at the chest of the user (see [6]), at a height of about 1.5 m from the floor. The height of the camera depends on the height of the user, though it is not relevant to the system's performance. The camera points forward, the image plane being almost vertical, and deviations due to swaying during walking are not problematic because problems are solved in the image processing steps.

When a user location is not certain due to missing or ambiguous visual landmarks, the user can take the camera from the mountable support and hold it in one hand for pointing it into different directions. Even a mobile phone with built-in camera can be used, as nowadays most models have a very good resolution.

Geographic information systems comprise tools for the processing, analysis, modeling, and storage of spatial data [10]. The growth of location-aware services and ubiquitous computing is leading to more indoor applications. In addition, since many buildings such as airports, shopping malls, and hospitals have become bigger and also more complex, more indoor GIS databases are becoming available, and increasingly more aging persons need more assistance in finding a destination [29].

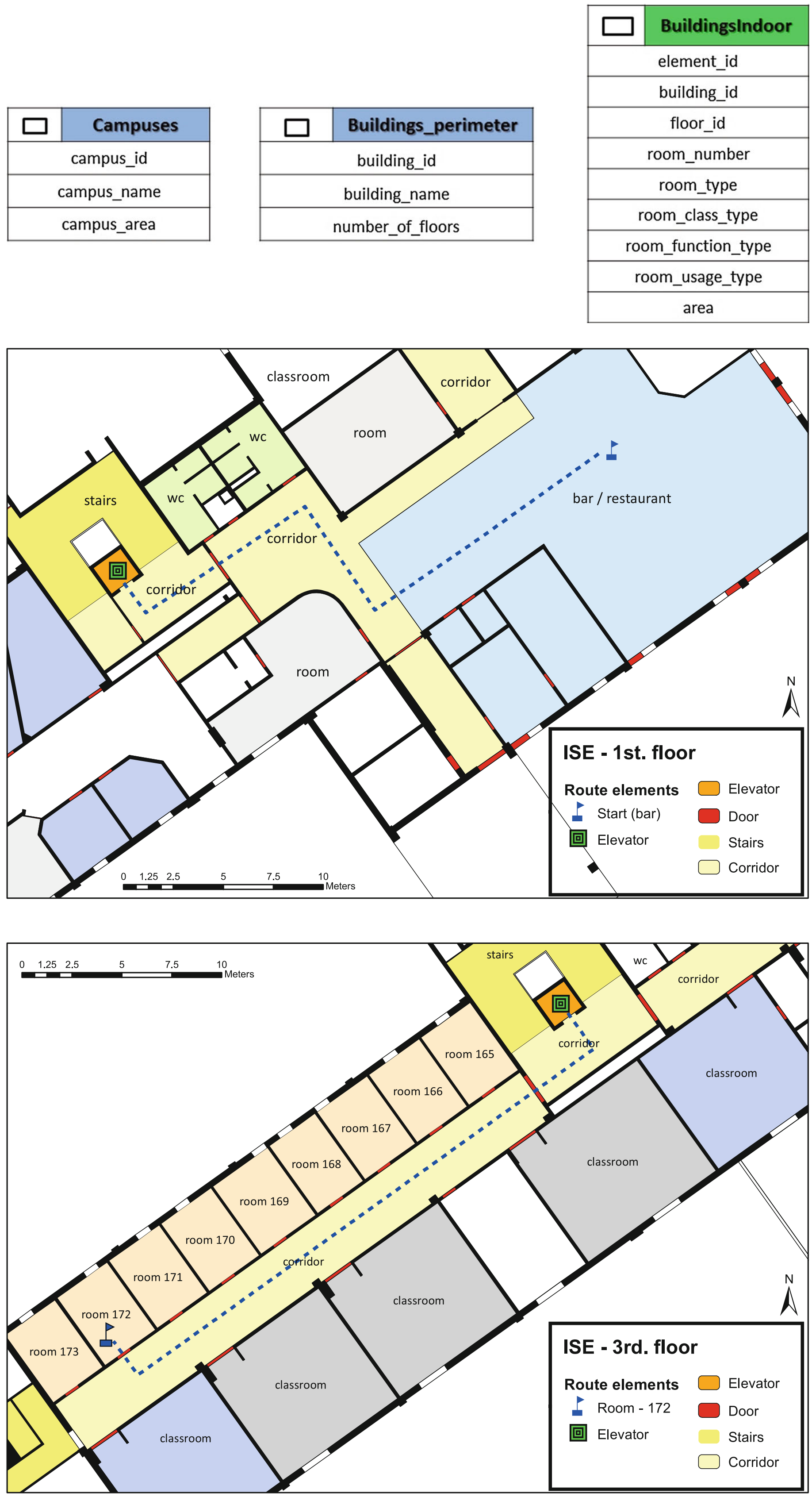

Fig. 1 The top three database levels (at *top*) and two partial maps of the ISE/UAlg building with two routes (*dashed*) on two floors, linked by an elevator

GIS data can be structured in raster and in vector formats. The GIS of ISE/UAlg employs vectorial data, and it was implemented using Postgresql/Postgis. The database model is illustrated in Fig. 1 (top). It includes three main classes *Campuses, BuildingsPerimeter*, and *BuildingsIndoor*. The datasets *Campuses* and *BuildingsPerimeter* are mainly for outdoor purposes: *Campuses* provides data about the areas and groups of buildings, whereas *BuildingsPerimeter* specifies the area and position of each building, its name, and the number of floors. *BuildingsIndoor* details all different spaces of each building. These include structural components, i.e., walls, doors and windows, circulation areas such as halls, corridors, stairs, elevators, and ramps, and rooms. The latter are classified by their type, function, and usage, for example office, classroom, lecture amphitheater, laboratory, library, and bar. The bottom part of Fig. 1 shows two partial maps of the 1st and 3rd floors of the ISE/UAlg building, including an example of a route which starts in the bar and goes to room number 172.

The system's component which handles spatial data was developed by using the *OGR Simple Features Library*. OGR is part of GDAL—the Geospatial Data Abstraction Library. OGR/GDAL is a project supported by the Open Source Geospatial Foundation (OSGeo) [9].

3 Landmark detection

Having information about the spatial layout of the building, it is necessary to discriminate all spaces. This can be done at a first level by localizing doors, windows, stairs, and elevators. However, this is not sufficient. All additional information which characterizes spaces, such as signs and objects, is useful for building a map, similar to simultaneous localization and mapping (SLAM) in robotics, e.g., [22].

Inspired by this idea, all useful landmarks such as objects, signs, and tags were retrieved and included in the GIS. Although anything can be stored in a GIS, it is not common to include things such as flowerpots, lockers, garbage bins, and vending machines. These may be important landmarks for localization, but they are not interesting for other GIS users. Hence, for a first prototype version, the more permanent landmarks were selected, such as fire extinguishers, exit, and WC signs, in order to complement doors, stairs, and elevators. Figure 2 (bottom) illustrates more examples. For detecting these special landmarks, the OpenSURF (Speeded-up Robust Features) library [8] was used. The SURF algorithm detects objects by finding correlations with templates. The attributes and points of interest of an object are invariant to scaling, rotation, distortion, and illumination. As a consequence, it is possible to identify objects almost regardless of their location, position, or rotation. Nevertheless, it is necessary to build a database of templates of each object as seen from several perspectives.

Although SURF is not computationally demanding, it is necessary to limit the search space of objects to small areas of interest instead of analyzing entire images. In the case of robotics, Nick's Machine Perception Toolbox [1] was

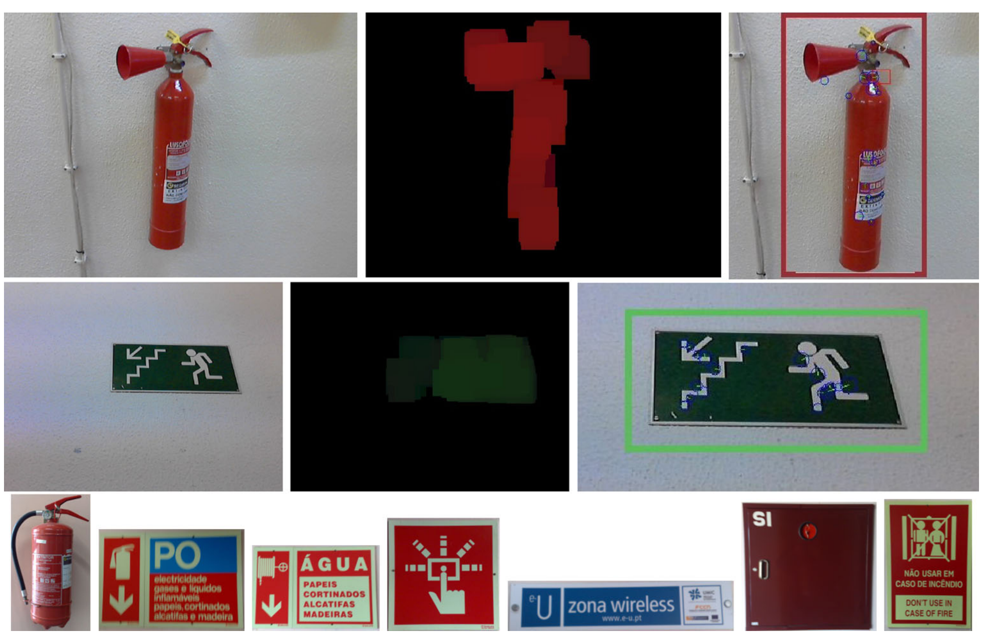

Fig. 2 *Top two rows, left to right* input images, color segmentations after dilations, and bounding boxes with SURF's interest points. *Bottom row* more examples of templates

applied for generating saliency maps. By thresholding these and by creating bounding boxes, region sizes can be limited to areas with possible objects [22]. In this case, the algorithm was simplified because (1) being an indoor space of a public building, most walls are white and even if they have a different color, they are normally light and homogeneous, and (2) most signs and selected objects have rather pure and saturated colors. For the above reasons, a simple segmentation by color in RGB space was applied, as described next.

In a preprocessing step, the color histogram and the dominant color of each object and sign are computed. Then, each image is segmented using these colors with intervals of ± 25 of the R, G, and B components. The segmented regions are dilated with 30 iterations, which eliminate gaps due to other colors. The minimum and maximum coordinates in x and y of the dilated regions are used as bounding boxes. Finally, to the regions delimited by the bounding boxes, the SURF algorithm is applied to characterize the object. The top two rows of Fig. 2 illustrate, from left to right, the processing steps.

3.1 Doors

Detection of doors is challenging because there are different types with different frames, also with different geometries if viewed non-orthogonally. Below, two situations are addressed: (a) the detection of lateral doors when the user is walking along a corridor, and (b) more or less frontal doors, when the user wants to identify a specific door which he is facing.

3.1.1 Detection of lateral doors

Assuming that the user walks along a corridor guided by the local navigation module, more or less centered toward the corridor's end, most doors are viewed laterally. This complicates their geometry, although three things can be assumed: Doors have vertical frames, their height is larger than their width, and they connect to the floor. The latter means that they can be found close to the corridor (path) borders.

In a previous paper, the authors have been presented path and obstacle detection [14]. The goal is to guide the user on walkable paths, i.e., the area where the user can walk safely. Examples are indoor corridors and outdoor sidewalks. Because of perspective projection, the left and right border and other parallel lines intersect at the vanishing point, and this point defines the horizon line. Following preprocessing using the Canny edge detector [2], an adapted version of the Hough transform [7] is applied to extract the left and right borders from the image. Since vertical camera alignment varies over time when the user walks, the height of the horizon line is computed dynamically, by averaging the values of the previous five frames. The area inside the left and right borders and the bottom line of the image is called the path window. For more details concerning path detection see [14].

Detection of lateral doors starts when the user enters a corridor, i.e., the corridor's two borders on the floor have been detected. As the user is guided to walk along the corridor, doors enter the camera's field of view at a large distance such that they are rather small. Depending on their distance, there may not be enough information to confirm their geometry. Therefore, they are called candidates which will be tracked while approaching them. As the distance decreases, more information becomes available and a detailed analysis can be applied to confirm if a candidate is actually a door or another rectangular structure. Tracking is important because in some frames a door may be occluded, for example by another person.

After the Canny edge detector ($\sigma = 1.0$, $T_{\mathrm{l}} = 0.25$, and $T_{\mathrm{h}} = 0.5$), the result of which is already available because of path detection, the first step is to extract long vertical edge segments. The system looks for doors on both sides if the horizontal distances between the position of the vanishing point, and the leftmost and rightmost point of the path window are bigger than 50 pixels. If one distance is smaller, the corresponding side will not be analyzed because it does not show sufficient information. In addition, vertical edges are only checked in the left and right regions from the image borders to 3/4 of the distance to the vanishing point; these are the light gray regions in Fig. 3 (top).

For detecting the edge segments, the positions of edge pixels are stored as nodes in a graph, but all edges in the path window are ignored. The graph is constructed by checking all edge pixels on image lines, from left to right and top to bottom. However, instead of analyzing all image lines, in a first step, only every tenth line is selected, starting at 2/3 of the image height. Considering a horizontal interval of ± 5 pixels to the left and to the right of each edge pixel, the next 10th line for edge pixels in these intervals, if an edge pixel is found, a new node is created and linked to the edge pixel on the above 10th line. Figure 3 (middle) illustrates the checked lines and the created nodes. Edge segments with less than three connected nodes can be discarded because they are not long enough.

The next step is to verify vertical edges between the sampled lines. The top and bottom nodes of each edge segment define a straight line where edges are checked. Starting at the middle of the line and going up and down until the end points, edge continuity is checked by applying a distance tolerance of one pixel to each side. Small gaps are filled by interpolating the nearest, confirmed edge positions. Vertical edges below the path borders are not

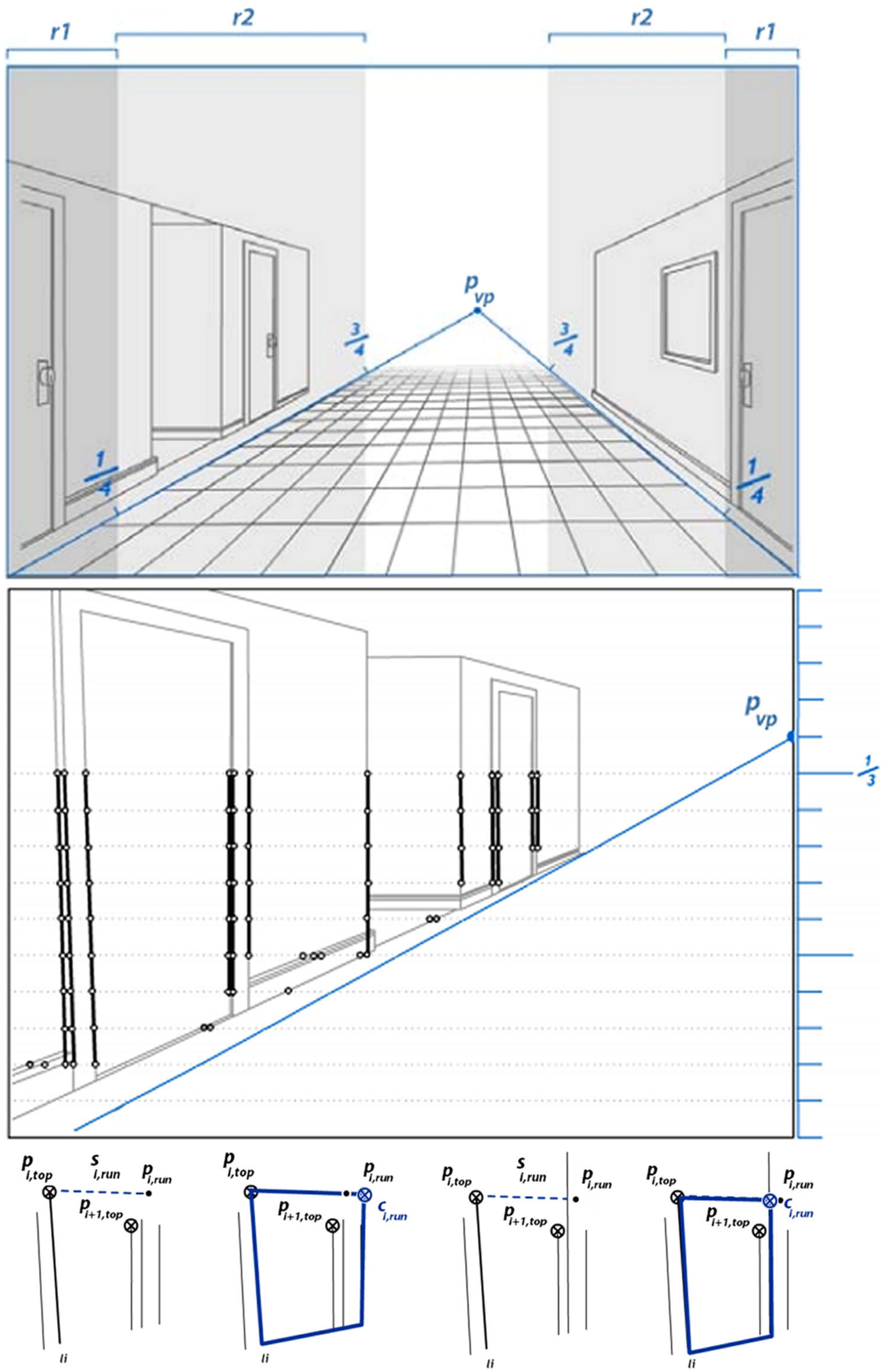

Fig. 3 *Top* at the *left* the regions in which doors are detected, and at the *right* part of the door detection algorithm. *Bottom* detection of quadrilaterals; see text

checked, but those above 2/3 of the image height are in order to obtain the most complete edge information. The result is a list of confirmed and significant vertical edges.

Since short vertical edges are not likely part of door frames, they are discarded. Taking into account perspective projection, the vertical distance between the path border on the floor and the corresponding (left or right) border of the ceiling can be easily computed because they pass through the vanishing point. This distance is a linear function of the distance to the camera. All vertical edges with a length shorter than 2/3 of their corresponding distance are ignored.

Having a list of long vertical edges, quadrilateral geometry must be checked for confirming door frames. To this end, the neighborhoods of the top points of the vertical edges are first checked for corners. The Harris corner detector [11] is applied with a window of size $CR_{size} = 5 \times 7$ pixels. This detector is based on the fact that corners cause strong partial derivatives. If the two eigenvalues of the so-called Harris matrix H are approximately similar and above a threshold which depends on the size of the window, there is likely a corner present. The corner response is based on the approximation $CR = \lambda_x \lambda_y \approx \det(H) - k \times \text{trace}(H)$, with $k = 0.8$.

Let $P_{i,\text{top}}$ be the top point of vertical edge i, and the local maximum of the corner detector be at position $P_{c,i,\text{top}}$. If CR at $P_{c,i,\text{top}}$ is higher than a threshold (10^{-6}), a corner is assumed, and $P_{i,\text{top}}$ is substituted by $P_{c,i,\text{top}}$. After all top points of vertical edges have been checked for corners, an edge-tracking procedure is started, from each corner point in the direction of the vanishing point, along the connecting Bresenham line, but with an angular tolerance of about five degrees. This means that the tracked edge is extrapolated, pixel by pixel, in the direction of the vanishing point, and the maximum vertical color gradient is sought (the sum of the differences of the R, G, and B values). The mean of the color gradient along the tracked part is also computed, from the start point $P_{c,i,\text{top}}$ to the current position. The tracking stops when the color gradient at a new position differs more than 40 from the mean over the tracked part. In that case, the tracking can still continue a few pixels further, using a new mean, and if successful, the small gap can be filled. If not successful, the tracking has stopped at a point $P_{i,\text{run}}$, and this forms, together with $P_{c,i,\text{top}}$, an edge segment $S_{i,\text{run}}$. A quadrilateral is formed by searching for the top of the vertical edge which is nearest to $P_{i,\text{run}}$; see Fig. 3 (bottom-left). If the nearest top position is above $P_{i,\text{run}}$, see Fig. 3 (bottom-right), the edge is trimmed to the intersection point. The same process is repeated for all i.

A quadrilateral found in one frame may not be detected in another frame. This can be due to excessive camera tilt, or because of poor or excessive lighting which can hide real edges or create shadow edges. For these reasons, quadrilaterals, once detected, are tracked in subsequent frames. The quadrilaterals' bottom points, close to the floor and therefore close to the path border, are good indicators for tracking. In principle, all bottom points are collinear, for example four points of two doors, and their distance ratios in two consecutive frames are preserved in perspective projection. Hence, tracking over frames could work if there are at least four matching edges in two consecutive frames. However, this is unlikely to happen, mainly because of perspective occlusion of the closest inner edge between door surface and door frame by the frame itself.

Therefore, the following approach was adopted: If a quadrilateral is found in region r_1, see Fig. 3 (top), the search for the quadrilateral in the next frame will begin in the same region of the quadrilateral's bottom points, considering a displacement of three pixels in both directions along the path border. If a corner of the quadrilateral enters region r_2, a template patch of the bottom point is saved for further localization. The same interval plus displacement is considered, but each bottom point of vertical edges is tested against the patch.

Figure 4 shows examples of corridors with detected path borders and doors in red. The top row shows door detection and tracking while approaching an obstacle in the center of the corridor. The bottom row shows more examples of door detection. In the right two frames, one can see that some

Fig. 4 *Top* path, obstacle, and doors detected along a corridor. *Bottom* more examples

doors have not been detected, because they are too close or too far. Nevertheless, these are detected in other frames due to the tracking, such that the user can be informed about their existence.

Detection of lateral doors was tested using video sequences captured while moving a camera along corridors, such that the doors could be tracked over time. The corridors were in three buildings, including ISE/UAlg, because of the different architectures: doors in the wall plane versus recessed in niches, wall, and door paints, periodic incandescent versus more continuous fluorescent illumination along the corridor, and tidy versus niches occupied by metal lockers. Most doors, about 80 %, could be detected and tracked. False positives were mainly due to vertical edges of real doors which were connected by a horizontal edge caused by wall paints. Unfortunately, other studies on the detection of lateral doors did not provide quantitative results; hence, a direct comparison of the results is not possible.

3.1.2 Frontal doors and their classification

For detecting frontal doors, two types of rectangle detectors are combined for increasing robustness: (a) The Hough transform is applied to edges detected by the Canny algorithm (the same as in the previous sections). After selecting vertical and horizontal edges, the intersections are detected and rectangular shapes are analyzed. (b) An unsupervised binary segmentation is checked for rectangular shapes. Both algorithms must detect the same rectangle before a candidate door is subjected to final verification.

In method (a), the Canny edge detector is applied to $I_i(x, y)$, the input frame. This results in I_b, to which the Hough transform is applied, yielding the edge histogram $I_H(\rho, \theta)$. Only (almost) horizontal and vertical edges are relevant, so the following orientation windows are applied: $70° < \theta < 110°$ for horizontal edges and $-20° < \theta < 20°$

Fig. 5 Frontal doors. *Top two rows, from left* input image, detected horizontal and vertical edges and intersection points, detected rectangles of method (a), plus segmented image, detected rectangles and selected ones of method (b), and final result. *Third and 4th rows* another door. *Bottom row* examples of texts extracted from doors

for vertical ones. To determine the intersections of two edges, the equations $\rho_1 = x\cos\theta_1 + y\sin\theta_1$ and $\rho_2 = x\cos\theta_2 + y\sin\theta_2$ are simply applied. All intersection points are grouped in sets that define rectangular shapes. The above processing steps related to method (a) are illustrated in Fig. 5: the top row shows, from left, I_i, I_b and all detected intersections and the detected rectangles, in this case belonging to a double door. The corresponding images on the 3rd row show results in the case of a partial view of a single door. The remaining images on the 2nd and 4th rows illustrate processing using method (b), see below.

Method (b) is based on K-means clustering with only two classes and a segmentation by using minimum distance pixel classification to the two class centers. The next step consists of searching for rectangles in the binary image, i.e., connected regions of pixels (black in this case). Rectangles extracted by this method are validated as door candidates if they are delimited by two vertical edges and a neighboring horizontal one. The images on the 2nd and 4th rows of Fig. 5 show segmentation results, detected rectangles, and selected ones.

Door candidates (DC) identified by the two methods are further screened in order to eliminate all rectangles which are not likely part of a door, assuming that doors have a height H greater than their width W and occupy a substantial area A of the image. All door candidates that do not satisfy the relations $A_{DC} \geq T_A$ and $(H_{DC}/W_{DC}) \geq T_R$ are ignored. The area threshold $T_A = 16~\%$ of the total number of pixels of the image and the ratio threshold $T_R = 1.2$.

After a door has been detected, it must still be recognized. In public buildings, most doors are identified by a sign or a text label. This is very important because (1) the user may want to go to a specific room, and (2) in corridors with several doors, the GIS can be used for localization.

Recognition of common signs on doors is solved by landmark recognition. For text recognition, "Tesseract-OCR" [27] is applied. This application is free for academic and commercial use. Tesseract accepts an input image and returns a text file with the OCR result. Although Tesseract has mechanisms for correcting spelling errors, there still remain errors in the form of fractions of invalid text. Hence, there still is a need for post-processing in order to obtain useful and valid information. The bottom row of Fig. 5 shows texts on doors in the ISE/UAlg building. For example, in case of the left two texts, Tesseract created the following two strings: "Escola Superiorde Tecno\og'\ a" and "-\ 160'i." Both strings are partly correct, but contain snippets of invalid text. This issue was addressed by developing an algorithm which isolates and corrects each word, using predefined words in a reference database. This database is created dynamically from the GIS information and consists of existing labels in the space where the user is. Normally, there are less than a few dozens of words.

The correction algorithm first eliminates all non-alphanumeric characters and then compares the remaining character string, starting with the first character or the first one after a detected and validated word, with all words in the database. If the number of matching characters is bigger than about 80 % of a word, the substring is substituted by the word in the database. In nearly all cases, including the texts on the bottom row of Fig. 5, the result is correct.

Detection of frontal doors was tested on four types: elevators, double and single doors, and glass doors with a frame. The tests involved real conditions, i.e., different viewpoints, light sources and levels, and occlusions; see the two examples in Fig. 5. On the set of 200 test images, a detection rate of 82 % was achieved. Tian et al. [28] achieved better results on a different dataset with a total of 203 images containing 210 doors in various environments (elevators, open doors, glass doors, and book cases; different colors and textures, viewpoints, light conditions, and occlusions; image resolution 320 × 240). Their detection rate was 92 % with a false-positive rate of 3 %.

3.2 Detection of stairs

Stairs consist of a series of steps with almost parallel edges, and the distance between the steps varies almost linearly. As a chest-mounted or handheld camera is used, edges may not be horizontally aligned. This variation in orientation of the camera is also responsible for the lines between steps not being completely parallel.

The first processing step is similar to the one in the detection of doors as explained before. Canny's edge algorithm and the Hough transform are applied; however, in this case, focus is on horizontal and almost horizontal lines for which $-35° < \theta < 35°$.

Horizontal and vertical surfaces of the steps may consist of differently colored or textured materials. In order to eliminate possible errors which are caused by the different materials, and to improve edge periodicity, it is necessary to apply a minimum threshold to the distances between the edges which mark parts of individual steps. After analyzing many different stairs, a minimum distance of 5 % of the frame height is applied. If the distance between two lines is bigger than the minimum distance, the edges are kept. All lines with a smaller distance are discarded and replaced by only one with the average vertical position and orientation of the discarded lines. For non-horizontal edges that have about the same orientation (±5°), the process is similar, but only the most horizontal ones are kept.

This preprocessing yields almost horizontal and periodic lines (edges) which can occur in the entire image. Keeping in mind that a blind person will walk with the white cane and that the goal is to inform the person on approaching a

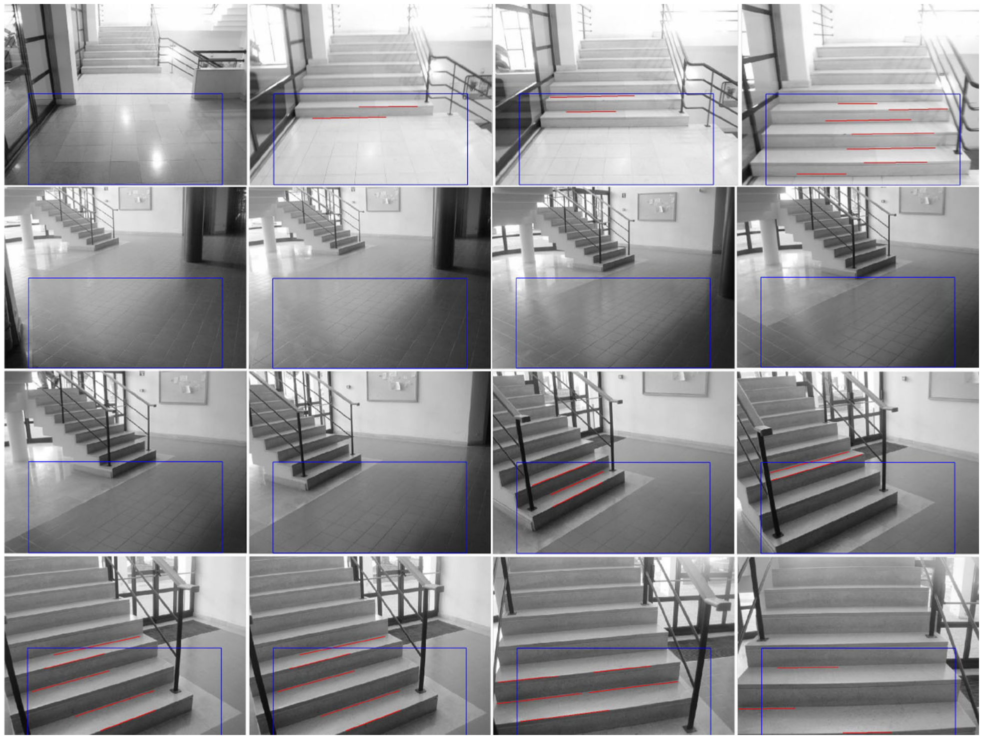

Fig. 6 Two sequences with stairs. The ROI is indicated by the *blue rectangle*. The *red lines* are parts of edges which passed the candidate test

staircase, a region-of-interest (ROI) is applied. The width and height of this ROI are 80 and 50 % of the image width and height, respectively, discarding the left, right, and top parts of the image. Candidate stairs are detected if (1) at least 2 lines exist in the ROI, (2) each of these has a length of at least 20 % of the ROI's width, and (3) these lines are a majority with congruent angles. The latter criterion serves to discard spurious lines with different angles.

When a candidate staircase has been detected, a final validation is applied. Since only edge information has been exploited in the previous steps, now pixel values are considered because most stairs appear as bright and dark horizontal bars. First, a vertical region with a width of 7 pixels in the center of the ROI is selected, and on each line, the 7 pixels are averaged to reduce noise. The resulting vector is then thresholded by using the central minimum of the histogram of all pixels in the vertical region (because this region has more pixels than the average vector). Validation is positive if there are at least 2 bright bars separated by a dark one, each bar consisting of at least 10 connected vertical pixels. Figure 6 shows two sequences containing stairs with different viewing angles. The ROI is indicated by the blue rectangle, and the red lines are (parts of) edges which passed the candidate test in case if the stairs also passed final validation.

It should be stressed that false positives cannot be avoided, because there are many quasi-periodic and "horizontal" structures, for example zebra crossings (outdoor), pavements which consist of differently colored tiles (in- and outdoor), and (book) shelves (indoor). The dataset used to test the algorithm contains images of many stairs captured at different sites, both in- and outdoor. Although this paper is about indoor navigation, the prototype also serves outdoor navigation and stairs; there are also an important hurdle. The set contains a large variety of styles (open and solid construction, different materials), views (oblique, frontal, seen from the top and bottom floors), and distances. It also contains many images with quasi-periodic patterns such as zebra crossings, pavements, and benches with shadows; see Fig. 7.

Of all images without stairs, only 7 % were detected as having stairs. This low false alarm rate is mainly due to insufficient periodicity of non-stairs patterns. Of all images containing stairs, 88.5 % were correctly detected. The false-positive rate of 11.5 % is mainly caused by insufficient lighting or low contrast due to the stairs' materials. Hence, it makes sense to repeat the experiments after applying brightness and contrast correction, for example histogram equalization, but a false-positive rate of 0 % is not very realistic. Once again, it should be stressed that the white cane must be used to check the space in front, but at least in most cases, the user will be informed about objects and problems before they are encountered by the swaying cane. The experimental

Fig. 7 Some examples of the stairs dataset; see text for detailed explanation

results were obtained without exploiting any context information. In case of indoor navigation (see Sect. 4), context information is available through GIS-based localization. In addition, temporal consistency over consecutive frames can be used together with estimation of the ground plane, although the latter implies stereo vision [15]. The latter authors aimed at reducing false positives, but presented results in terms of ROC curves with recall and precision statistics (they mention 501 false positives in 852 frames which contain multiple stairs). To the best of authors knowledge, all other studies on the detection of stairs using a monocular camera did not provide quantitative results; hence, a direct comparison of the results is not possible.

4 Navigation

Being able to detect doors, stairs, and many objects and signs, and by including their positions in the database, GIS can be used for route planning and user localization [25]. These are described below.

4.1 Route planning and tracking

As already mentioned, maps based on the GIS of the ISE/UAlg building can be used for user navigation. Data from the GIS system are retrieved by using GDAL (see Sect. 2). GDAL allows to extract all information concerning geographical divisions (spaces, rooms, corridors), landmarks, and structures such as walls, doors, and windows. The top left map in Fig. 8 shows one room highlighted in red, and the table next to it lists additional information of that room. It is also possible to integrate specific landmarks such as objects and signs at different GIS layers. All information can be used to create a map suitable for navigation. For example, in division X, there are four walls, one window in the wall opposite to the door, a fire extinguisher at the left side of the door, an exit sign to the right, etc. In addition, it is straightforward to determine a representative location of each room on the basis of the geometric centroid. Figure 8, top right, illustrates two locations by the green and red dots. This example will be used to illustrate path planning, where these dots are initial start and end points (during navigation the user can indicate a new destination).

Specifically for navigation, a data structure was created. This relates data of, for example, a room with its neighboring spaces: Room X has one door which connects to room W, and another door which connects to corridor V. In turn, room W has only one door which connects to room X, and so on. These relations are very important and useful for path planning between start and end points on a same floor. If a desired destination is on another floor, as illustrated in

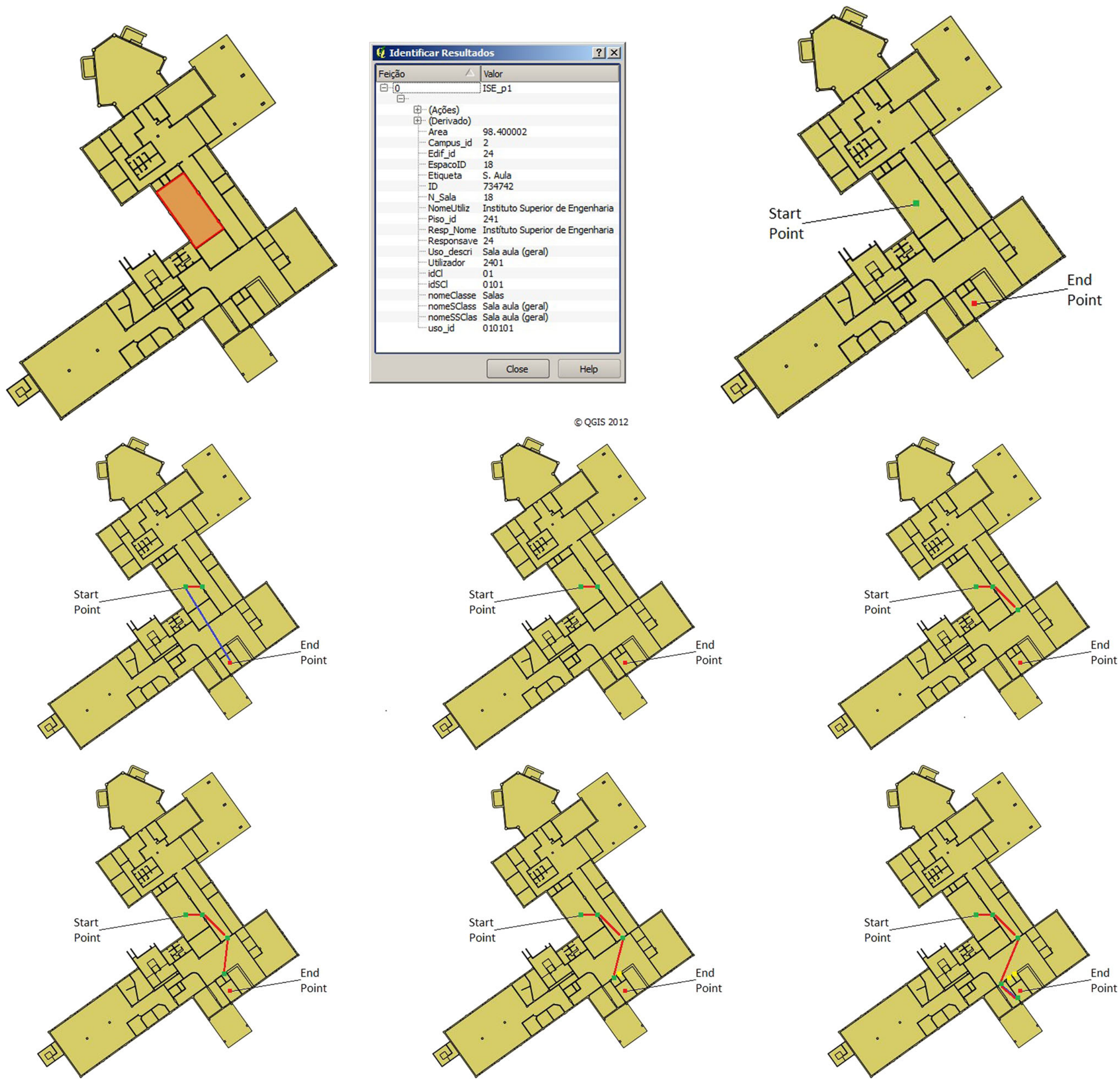

Fig. 8 *Left to right* and *top to bottom* 1st floor of the ISE/UAlg building and step-by-step route planning; see text for explanation

Fig. 1, the closest elevator or staircase on the actual floor will be selected as end point, and the elevator or staircase will be a new start point on the new floor.

Figure 8 illustrates, from left to right and top to bottom, the 1st floor of ISE/UAlg with all the steps for building a valid route between the start and end positions, i.e., the green and red dots. Step-by-step, the straight routes (blue lines) are substituted by real routes (red lines), first through an existing door, then to the end of the corridor, across the hall, to the entrance of the other corridor, etc. This process is repeated until the end point. Note that the two yellow dots in Fig. 8 mark doors of spaces without access of normal persons (electric installation, air conditioning), and these are therefore omitted.

4.2 Localization during navigation

Once a valid route has been built, the special data structure for path planning, which also contains the final path, can be integrated with the GIS. The latter holds, apart from all divisions etc., all registered landmarks: doors, stairs, elevators, and all kinds of signs. In other words, the system knows where the user is *assumed* to walk, but it cannot be sure about the actual location. In addition, the user can change his mind and decide to go to another destination.

The vision system is always active, analyzing incoming image frames for landmarks. Every time it detects and recognizes a landmark, the landmark type is checked in the GIS at the current location, i.e., the assumed space or division. If its type can be confirmed, the current location is also confirmed. If not, then the neighboring divisions are tested by using the last part of the traced route. In case of doubts, the user is instructed to point the chest-mounted or handheld camera slowly into different directions while standing or slowly walking about.

A division is recognized and confirmed if at least three visual landmarks can be confirmed in the GIS. If less than three have been confirmed, all neighboring spaces with those landmarks are checked and memorized as likely locations, until a third landmark is encountered. In the case that absolute certainty cannot be obtained, the region formed by the (assumed) actual and neighboring spaces is enlarged by applying an increasing circular area in the GIS and checking the most recently detected landmarks in this area. Also, most likely errors are checked, for example an elevator detected as a normal door, but represented as an elevator in the database.

5 Conclusions

This paper presented a system which integrates an indoor GIS of a building with visual landmarks detected by a normal, chest-mounted, or handheld camera. It serves to improve user autonomy in finding a destination. Visual landmarks are used to localize the user, and routes to the destination are dynamically updated by tracing detected landmarks. The system was designed such that it can be integrated in the "Blavigator" prototype. The latter is able to detect valid paths and any obstacles for local navigation [14, 17]. The system presented here complements local navigation with global navigation, but only indoor. In addition, detection of obstacles on the ground in front of the user is complemented by detection of doors and stairs and other objects, such that the user can build a more complete mental map of the building.

The system works in real time on a netbook computer, and it was tested successfully in the ISE/UAlg building, first by sighted persons and then by blindfolded ones. Most planned routes could be followed from the start to the destination, even when these were on different floors. The few which caused a problem were due to a lack of landmarks at a certain location. However, after finding a new start position by walking about to a location with multiple landmarks, the updated routes could also be accomplished.

Regarding the detection of doors, both lateral doors and frontal-facing ones were considered. The first show good detection results when they are at a range of about five meters, problems mainly being due to occlusions, low contrast, and wall patterns. Frontal-facing doors also show good detection and classification rates, although text recognition must be improved. In the case of stairs, despite the excellent results, errors still occur when the lighting is not sufficient and tile or paint patterns on the floor and walls are very similar to those of stairs.

Ongoing work concerns the following: (1) including more landmarks in the database such that all spaces can be covered more densely, (2) improving the detection of doors and stairs especially at oblique viewing angles, such that the user can limit the pointing angles of the camera to $\pm 45°$ from the front, and (3) developing a speech-based interface with queries and messages which replaces the provisional one based on different beeps. The final goal is a system which only employs a smartphone with a built-in camera, worn by a strap around the neck. Tests with blind persons, in collaboration with ACAPO, the Portuguese association of blind and amblyopes, are already planned.

Acknowledgments This work was partly supported by the Portuguese Foundation for Science and Technology (FCT), project PEst-OE/EEI/LA0009/2013 and project Blavigator RIPD/ADA/109690/2009.

References

1. Butko, N., Zhang, L., Cottrell, G., Movellan, J.: Visual salience model for robot cameras. In: Proceedings of the IEEE International Conference on Robotics and Automation, pp. 2398–2403 (2008)
2. Canny, J.: A computational approach to edge detection. IEEE Trans. Pattern Anal. Mach. Intell. **8**(6), 679–698 (1986)
3. CASBliP: Final activity report of the EU-funded CASBliP project (2009). http://casblipdif.webs.upv.es
4. Chen, Z., Birchfield, S.: Visual detection of lintel-occluded doors from a single image. In: Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops, pp. 1–8 (2008). doi:10.1109/CVPRW.2008.4563142
5. Costa, P., Fernandes, H., Vasconcelos, V., Coelho, P., Barroso, J., Hadjileontiadis, L.: Landmarks detection to assist the navigation of visually impaired people. In: Proceedings of the 14th International Conference on Human–Computer Interaction: Towards Mobile and Intelligent Interaction Environments, vol. 3, pp. 293–300 (2011)
6. du Buf, J., Barroso, J., Rodrigues, J., Paredes, H., Farrajota, M., Fernandes, H., José, J., Teixeira, V., Saleiro, M.: The SmartVision navigation prototype for blind users. JDCTA Int. J. Digit. Content Technol. Appl. **5**(5), 351–361 (2011)
7. Duda, R., Hart, P.: Use of the Hough transform to detect lines and curves in pictures. Commun. ACM **15**, 11–15 (1972)
8. Evans, C.: Notes on the OpenSURF Library. Technical report CSTR-09-001, University of Bristol (2009). http://www.chrisevansdev.com
9. GDAL Development Team: GDAL—geospatial data abstraction library, version 1.9.0, open source geospatial foundation (2011). http://www.gdal.org
10. Goodchild, M.: Geographic information systems and science: today and tomorrow. Proc. Int. Conf. Min. Sci. Technol. **1**(1), 1037–1043 (2009)

11. Harris, C., Stephens, M.: A combined corner and edge detector. In: Proceedings of the 4th Alvey Vision Conference, pp. 147–151 (1988)
12. Hernandez, D., Kang-Hyun, J.: Outdoor stairway segmentation using vertical vanishing point and directional filter. In: Proceedings of the International Forum on Strategic Technology, pp. 82–86 (2010)
13. Hernandez, D., Kim, T., Jo, K.: Stairway detection based on single camera by motion stereo. In: Proceedings of Modern Approaches in Applied Intelligence—24th International Conference on Industrial Engineering and Other Applications of Applied Intelligent Systems, Syracuse, NY, USA, Springer LNCS, vol. 6703, pp. 338–347 (2011)
14. José, J., Rodrigues, J., du Buf, J.: Visual navigation for the blind: path and obstacle detection. In: Proceedings of the International Conference on Pattern Recognition Applications and Methods, vol. 2, pp. 515–519 (2012)
15. Lee, Y., Leung, T., Medioni, G.: Real-time staircase detection from a wearable stereo system. In: Proceedings of the 21st International Conference on Pattern Recognition, Tsukuba, Japan, pp. 3770–3773 (2012)
16. Lu, X., Manduchi, R.: Detection and localization of curbs and stairways using stereo vision. In: Proceedings of the IEEE International Conference on Robotics and Automation, pp. 4648–4654 (2005)
17. Moreno, M., Shahrabadi, S., José, J., du Buf, J., Rodrigues, J.: Realtime local navigation for the blind: detection of lateral doors and sound interface. In: Proceedings of the 4th International Conference on Software Development for Enhancing Accessibility and Fighting Info-exclusion, Porto, Portugal pp. 74–82 (2012)
18. Munoz-Salinas, R., Aguirre, E.,Garca-Silvente, M., Gonzalez, A.: Door-detection using computer vision and fuzzy logic. In: Proceedings of the 6th WSEAS International Conference on Mathematical Methods and Computational Techniques in Electrical Engineering, p. 6 (2004)
19. Murillo, A., Kosecka, J., Guerrero, J., Sagues, C.: Visual door detection integrating appearance and shape cues. Rob. Auton. Syst. **56**(6), 512–521 (2008)
20. Ran, L., Helal, S., Moore, S.: Drishti: an integrated indoor/outdoor blind navigation system and service. Proc. Second IEEE Annu. Conf. Pervasive Comput. Commun. **11**(6), 23–30 (2004)
21. Roth, P., Winter, M.: Survey of appearance-based methods for object recognition. Technical report ICG-TR-01/08, Institute for Computer Graphics and Vision, Graz University of Technology, Austria (2008)
22. Saleiro, M., Rodrigues, J., du Buf, J.: Minimalistic vision-based cognitive SLAM. In: Proceedings of the 4th International Conference on Agents and Artificial Intelligence, Special Session Intelligent Robotics (ICAART-SSIR2012), Vilamoura, Portugal 1, pp. 614–623 (2012)
23. Schmitz, B., Becker, S., Blessing, A., Matthias, G.: Acquisition and presentation of diverse spatial context data for blind navigation. In: Proceedings of the IEEE 12th International Conference on Mobile Data Management, vol. 1, pp. 276–284 (2011)
24. Se, S., Brady, M.: Vision-based detection of stair-cases. In: Proceedings of the Asian Conference on Computer Vision, pp. 535–570 (2000)
25. Serrão, M., Rodrigues, J., Rodrigues, J., du Buf, J.: Indoor localization and navigation for blind persons using visual landmarks and a GIS. In: Proceedings of the 4th International Conference on Software Development for Enhancing Accessibility and Fighting Info-exclusion, Porto, Portugal, pp. 65–73 (2012)
26. Singh, V., Paul, R., Mehra, D., Gupta, A., Sharma, V., Jain, S., Agarwal, C., Garg, A., Gujral, S., Balakrishnan, M., Paul, K., Rao, P., Manocha, D.: "Smart" cane for the visually impaired: design and controlled field testing of an affordable obstacle detection system. In: Proceedings of the 12th International Conference on Mobility and Transport for Elderly and Disabled Persons, Hong Kong, China, p. 13 (2010)
27. Tesseract-ocr (2011). http://code.google.com/p/tesseract-ocr/
28. Tian, Y., Yang, X., Arditi, A.: Computer vision-based door detection for accessibility of unfamiliar environments to blind persons. In: Proceedings of the 12th International Conference on Computers Helping People with Special Needs, Springer LNCS, vol. 6180, pp. 263–270 (2010)
29. Worboys, M.: Modeling indoor space. In: Proceedings of the 3rd ACM SIGSPATIAL International Workshop on Indoor Spatial Awareness, pp. 1–6 (2011)
30. World Health Organization: Visual impairment and blindness. Fact sheet 282, October 2011. http://www.who.int/mediacentre/factsheets/fs282/
31. Zhong, C., Zhuang, Y., Wang, W.: Stairway detection using Gabor filter and FFPG. In: Proceedings of the International Conference of Soft Computing and Pattern Recognition, pp. 578–582 (2011)