J Multimodal User Interfaces (2015) 9:287–297
DOI 10.1007/s12193-015-0191-6

ORIGINAL PAPER

Blind guide

A virtual eye for guiding indoor and outdoor movement

Bálint Sövény[1] · **Gábor Kovács**[1] · **Zsolt T. Kardkovács**[1]

Received: 14 December 2014 / Accepted: 13 July 2015 / Published online: 11 August 2015

Abstract In this paper, we present a design of a wearable equipment that helps with the perception of the environment for visually impaired people in both indoor and outdoor mobility and navigation. Our prototype can detect and identify traffic situations such as street crossings, traffic lamps, cars, cyclists, other people and obstacles hanging down from above or placed on the ground. The detection takes place in real time based on input data of sensors and cameras, the mobility of the user is aided with audio signals.

Keywords Virtual eye · Stereo vision · Augmented reality

1 Introduction

Visually impaired people face difficulties in everyday traffic. A white rod can detect certain obstacles on the ground level in a close proximity, there are, however, a lot more traffic situations its user should be aware of: quickly moving objects, public transportation, hanging objects, traffic lamps without audio aid, or the speed of other passerbys. Guide dogs are trained for these purposes, but that is not the ideal solution as the training is expensive and time consuming, hence there is a need for a simple, quickly deployable and language independent tool. According to WHO there are about 39 million blind people and another 246 million visually impaired, so the potential market for such a tool is about 3 % of the population of the world.

1.1 Our contribution

In this article, we propose a virtual eye tool that utilizes the experience of the cognitive aid system for blind people (CASBliP) projects regarding practical requirement of blind people. Our prototype tool (see Sect. 6) consists of a camera mounted on a helmet, its image is processed with a portable computer carried by the user in real time. Earlier research prototypes were limited to 32 × 32 images [18] and depth maps of similar size [2], which resulted in heavy information loss for instance traffic lamp detection impossible. In our work, we use full scale images (1920 × 1080 for recognition, 640 × 480 for depth map), and yet process them automatically in real time. Our image processing algorithms cover the classification and the risk analysis of hanging obstacles, crosswalks, stairs, traffic lamps, which is much a broader spectrum than what previous virtual eye solutions could provide, and are able to find doors and space between passerbys.

Next in this section, we present a short outline of the relevant projects. The rest of the article is organized as follows. In Sect. 2, we give the high level overview of the blind guide prototype and the main tasks of each functional module. We describe the image transformation and object detection algorithms we use in Sects. 3 and 4 respectively. Section 5 describes the threat assessment and the threat to audio signal conversion. Section 6 shows the experience of tests and public demonstration. Finally, we give a brief summary in Sect. 7.

This article is an extended version of the paper that appeared at CogInfocom 2014.

✉ Gábor Kovács
kovacsg@u1research.org

Bálint Sövény
soveny@u1research.org

Zsolt T. Kardkovács
kardkovacs@u1research.org

[1] U1 Research Ltd., Gabor Denes 2, Budapest, Hungary

1.2 Related work

Projects and tools targeting blind people have three main strategies. One is the replacement of the white rod with an electronic one, most projects focus on navigation how visually impaired people can get from one point of a city to another, and thirdly several research projects aim at developing a virtual eye using optical sensors to see and to be aware of the surrounding environment. In this section, we are focusing on the latter only. Blind guide solutions in a broad sense focus on the following issues, and they can be compared accordingly:

1. optical detection of obstacles
   (a) types of obstacles to be identified
   (b) range of the detection area
   (c) distance estimation and its derivatives (speed, acceleration)
   (d) occlusion and partial occlusion
   (e) environmental issues including indoor and outdoor specific issues
   (f) adaptive calibration
2. scenario analysis
   (a) self-trajectory estimation
   (b) multiple target trajectory estimation
   (c) threat estimation and danger evaluation
3. computational complexities
4. hardware problems
   (a) power consumption
   (b) degree of parallelism
5. human–computer interaction
   (a) non-visual representation of the surrounding environment
   (b) response time
   (c) ergonomy
6. wearability
   (a) physical dimensions
   (b) weight
   (c) design.

The two most influential projects aim at virtual eye may be the CASBliP and assisting personal guidance system for people with visual impairment (ARGUS), both funded by EU grants. They have not produced a commercial product yet, their experiences, however, need to be considered.

The EU funded CASPBliP [3] has made a useful survey based on personal interviews with blind people on their most important needs in navigation. The most critical ones are the following in descending order of priority that means the percentage of visually impaired people who marked the objects critical:

- poles: 55.02 %
- holes on the ground: 47.6 %
- vehicles: 42.36 %
- stairs: 34.93 %
- other people: 23.58 %
- hanging sunshade: 17.03 %
- traffic lamps: 13.1 %
- crosswalks: 11.97 %.

The survey addressed the minimum distance for detecting such objects as well. People, vehicles and holes on the ground should be detected from 20 m away, for the rest of the objects a detection distance of 5 m would be sufficient. In the project, stereo camera system or infrared sensors were used for 3D reconstruction and object detection, the camera movement is compensated with accelerometer sensor data. An integrated GPS device is used for navigation, and acoustic or vibration feedbacks are used for signaling. In the acoustic signals the volume represents the distance, the balance represents the direction, the frequency represents the the height of the object, and the sound pattern represents the object type. Physical dimensions, processing time, and power consumption data are not public: there are some demonstration videos available online.

The ARGUS [1] project was funded between 2011 and 2014 by the EU. That project focused on supporting blind people in autonomous movement and navigation. They use primarily GPS for navigation and make use of other radio frequency technologies such as WiFi, RFID and NFC in urban environment. ARGUS is focusing on georeferred areas, mainly in outdoor environments. The feedback about the 3D environment is provided with continuous signals on the acoustic interface and a tactile interface.

Some projects are using mounted optosensors on carried devices like an electronic white rod [6], a walking aid [13], or an electronic cane [12]. In these projects cameras of a mobile devices were used for detecting objects. Gude et al. [6] reads QR code labels on objects to identify specific locations. That tool has a detection range of 2.5 m and an identification range of 1 m. The project intended to provide feedback to the user on a Braille device. Kulyukin et al. [12] is using Microsoft Speech API for both control and feedback which cannot be used in crowded areas.

Several prototypes have been developed for aiding indoor and outdoor navigation of blind people. The RFID based solution in [5] by Ding et al. has good preliminary result in navigation, they are, however, short in range that results in high deployment cost, and the maintenance of the database that stores information about the environment is also an issue. The solution in [9] by Hub et al. uses WiFi signal strength specifically for indoor navigation. The user can control the

systems with voice commands or point to an object with a finger. In the latter case, the object is identified based on the picture of the mounted camera, and the name of the object is synthesized by a speech generator.

Client–server architecture based solutions are proposed in [7,11,16], where the camera and GPS data are sent from the client device mounted on the user to a remote operator who interprets the information and provides audio or tactile feedback to the user. A drawback of such solutions is that users may have to stop to understand the information provided by the operators, which weakens the ability of processing environmental sounds in the meantime. The object detection accuracy of such systems is a few meters.

In [14,15], the GPS based navigation is extended with a local database in a wearable computer that contains information such as points of interests and obstacles on the current location of the user. Three sensors are mounted on the user: the location is determined based on GPS signals, the orientation is based on a compass, and the movement speed is estimated based on accelerometer data. These data items are processed by a small portable device that contains all necessary information about locations including POIs. The system can be controlled with a keyboard or with speech commands. The feedback is provided via synthesized speech such that the intensity increases as the user gets closer to the target object. With regard to the large database, this solution is practical only in small sites.

A more advanced database based navigation system called Drishti is proposed in [17] by Ran et al. Outdoors, the proposed system uses GPS that can be switched to an ultrasound imaging based indoor navigation with a voice command. The wearable processing unit connects to a server via wireless connection, where localization is performed based on a database. The sensors are mounted on the shoulders of the user, hence there is no information about the height of the user, so the navigation is only two dimensional.

Heijden and Regtien [8] proposed the architecture of a wearable blind assistant that can be used for both indoor and outdoor navigation. The input data are collected by two cameras mounted on the glasses and and acoustic sensor mounted on the shoes of the user. The cameras are used for the reconstruction of the depth map if the movement of the cameras is known, and the acoustic sensors are used for detecting objects directly in front of the user.

Virtual eye based solutions are intended to detect and classify objects within the detection range of sensors and cameras mounted on the user. Kanna et al. [10] developed an FPGA based virtual eye device that detects objects based on the information collected by laser and infrared sensors mounted on the leg of the users, and classifies them based on an on-line image catalogue. Their device scans the predicted path with laser and infrared sensors mounted on the feet of the user to detect obstacles, the sensors connect to the central unit via ZIGBEE. The pictures of the cameras mounted on the head of the user are classified based on an on-line picture catalogue. Navigation is based on GPS. The user can give voice commands to the system, and gets feedback through the vibrating belts attached to the feet, which is practical because it does not suppress environmental noise.

Rao et al. [18] use pictures of wearable cameras as the input of their real-time virtual eye solution. The images are processed with a Texas Instruments Da Vinci Board such that first the objects are detected, then a threat level is assigned to each object, and an audio signal is generated based on the object with the highest thread level. The computational requirements are decreased by downscaling the image to 32×32 pixels. Objects are identified by edge detection on each color channel, threat level classification is based on the size and the position of the object; only objects in the direct foreground section of the image get the highest threat level.

In [2], Balakrishnan et al. use stereoscopic images to calculate a low resolution depth map analytically. The signaling is acoustic, where musical patterns with frequencies from frequency domain of the the human voice are generated based on the disparity of objects.

A short comparison of related projects are presented in Table 1. In the the table, we have used + whenever the solution uses a certain tool or provides a specific feature. There are too many types of obstacles to deal with, so in the types of obstacles column of the table ++ denotes that the solution can identify almost all relevant types determined by CASBliP project, + denotes that those solutions can make a difference between different obstacles without trying to identify all types, and − means no distinction is being made between different types. In the response time column, ++ stands for a response time in a well-defined target environment though not necessarily real-life conditions that allows a speed greater than 1.5 m/s, which is the walking speed of non-impaired people, + stands for the 0.65–1 m/s speed limitation, one step per second by non-impaired people, and − stands for a maximum allowed walking speed slower than 0.65 m/s.

2 The overview of the blind guide tool

The prototype tool has three main components: a stereo image acquisition module, a data processing unit (DPU), and a signaling generator unit. In the following, we give a brief overview of these.

Figure 1 shows the schematics of the stereo image acquisition unit. It contains two cameras that are deployed on both sides of the head symmetrically heading forward for supporting peripheral view. The field of view of the two cameras are not required to completely overlap, however, dead space must be avoided. Vertically, it is not required to mount the cameras at the same height. In our tool, we fixed the cameras on

Table 1 Comparison of different blind guide approaches

Solution	Types of obstacles	Distance estimation	Distance limitation (in m)	Threat analysis	Indoor	Outdoor	Audio map	Sonar map	Audio icon	Tactile
CASBliP [3]	++	+	20	−	−	+	−	−	+	−
ARGUS [1]	−	−	20	+	+	−	−	+	+	−
Gude et al. [6]	+	−	2.5	−	−	+	−	−	−	+
Guido [13]	−	−	20	−	+	−	−	−	−	+
e-Cane [12]	+	−	2	−	+	+	−	−	+	−
Ding et al. [5]	−	−	2	−	+	−	−	−	−	−
Hub et al. [9]	−	−	2	−	+	−	−	−	−	−
TUGS [7]	−	−	20	−	−	+	−	−	−	+
Loomis et al. [14]	−	−	20	−	−	+	−	−	−	−
Ran et al. [17]	+	−	5	+	−	+	−	−	−	−
Heijden and Regtien [8]	+	+	5	−	+	+	−	−	−	+
Kanna et al. [10]	+	−	2	−	+	+	−	−	−	−
Rao et al. [18]	++	−	10	−	+	+	−	+	−	−
Balakrishnan et al. [2]	++	+	10	+	+	+	−	−	+	−
U1 blind guide	++	+	10	+	+	+	+	−	−	−

Solution	Vibration	Speech	Braille	Response time	GPS	RFID/Wifi/NFC	Accelerometer	Gyroscope	Ultrasound	QR-codes
CASBliP [3]	+	−	−	−	+	−	+	+	−	−
ARGUS [1]	−	−	+	+	+	+	+	+	+	−
Gude et al. [6]	−	−	+	+	−	−	−	−	+	+
Guido [13]	−	−	−	++	−	+	−	−	−	−
e-Cane [12]	−	+	−	−	+	+	−	−	−	−
Ding et al. [5]	−	+	−	−	−	+	−	−	−	−
Hub et al. [9]	−	+	−	−	−	−	−	−	−	−
TUGS [7]	−	+	−	++	+	−	−	−	−	−
Loomis et al. [14]	−	+	−	−	+	−	+	+	−	−
Ran et al. [17]	−	+	−	+	+	−	−	−	+	−
Heijden and Regtien [8]	−	−	−	−	−	−	+	+	+	−
Kanna et al. [10]	+	+	−	++	+	+	−	−	−	−
Rao et al. [18]	−	−	−	++	−	−	−	−	−	−
Balakrishnan et al. [2]	−	+	−	−	−	−	−	−	−	−
U1 blind guide	−	−	−	+	−	−	−	−	−	−

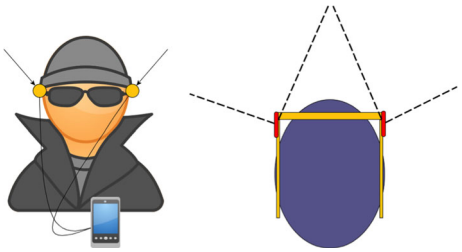

Fig. 1 Image acquisition setup for head mounted cameras

a helmet. By using a helmet for camera deployment, we can keep hands and feet of the users free, have a better line of sight, and the occlusion caused by hands is not a problem, however, we must compensate for the head movement.

The lines of sight of the cameras are directed a bit downwards so that both cameras can see the ground level directly the feet of the user, that is, only the next step which is about 30–50 cm is not covered. The horizon is in such cases closer to the upper edge of the camera pictures, and background objects above the horizon such as poles and houses can still be detected.

The data connection and the power supply of the cameras are provided by USB cables connected to the DPU. The DPU is a mini PC (see Sect. 6 for details) operated from an external power supply with a capacity enough for two hours' data processing. Functionally, the DPU consists of three modules: a data preprocessing unit, an image processing unit and a data evaluation unit. All of these three units are implemented in pure software.

The data preprocessing unit is responsible for performing necessary tasks that cannot be parallelized. Such tasks are the decoding and the transformation of data extracted from the cameras, and this is the module where the the rectification of images and color space transformation take place.

The image processing unit performs several tasks simultaneously: it searches for moving objects in the stereo image, it searches for crosswalks and stairs, and it searches for traffic lamps.

The data evaluation unit evaluates the traffic situation based on the outputs of the image processing unit. It determines the set of objects relevant with regard to the current trajectory of the user and the object, and it assigns a threat level to each object. The location, size and movement speed objects with a threat level above a threshold are passed to the signaling module.

The signaling generator unit generates an audio sound in the earphones. It must be noted that the volume of the sound is well below the environmental level, but still sensible, so that it does not deprive blind people from their most important sense. The tune identifies the object type (e.g. a hanging obstacle, stairs down), the volume the size or speed and the balance the direction relative to the symmetry line of the cameras' field of view.

Fig. 2 The image preprocessing workflow

3 Image preprocessing

The main tasks of image preprocessing are: image acquisition, rectification and image stabilization (see Fig. 2). The input of the preprocessing stage is the raw data extracted from the camera, the outputs are the rectified image, and a motion vector that compensates for the radial movements of the camera.

The preprocessing has to be performed in an uncertain environment like outdoor traffic. When designing image processing algorithm we have to take into account that

- light conditions are inconsistent, may change quickly within the angle of view of the camera as well, so color based object segmentation techniques [4] cannot be used,
- the fisheye optics that allows us to see far have high distortions,
- when estimating real distances we can use that the camera moves with about a 1–2 m/s speed, so with a recording speed of 25 fps, a man moves 4–20 cm and a vehicle moves up to 4 m between successive images, and
- the side movements of walking needs to be compensated.

3.1 Calibration

As in any stereo machine vision systems, camera calibration and a stereo calibration have to be performed for each deployment of cameras (see [4] for details). The pair of images extracted from the pinhole cameras have an inherent distortion that can be seen near the borders of the captured image. This causes straight lines on the sides and on the top to bend, in our context this means that straight poles can be seen as curves and there are no parallel lines, hence crosswalks and stairs cannot be detected. The distortion depends on the lens' characteristics and the deployment of the cameras. We can see in Fig. 3 that the buildings are not quite straight.

For each mount on a helmet, calibration has to be performed once, before the first usage. The output of the stereo calibration procedure is a couple of matrices (camera, distor-

Fig. 3 Images extracted from the cameras

Fig. 4 The undistorted camera pictures

tion, rectification and projection), with which it is possible to define a one-to-one correspondence between the pixels of the two images captured at the same time by the two cameras. The stereo calibrated and undistorted image pair is shown in Fig. 4.

3.2 Stabilization

The other problem of image acquisition is that the camera itself moves, and we have to compensate for the small radial and up-and-down movements caused by human walk to get a stable image and to be able to calculate the motion vectors. This compensation is performed before any object detection takes place, otherwise it cannot be determined if an object moved in the inertial reference of the camera or the displacement was caused by the small movement of the camera.

For stabilization, we use sparse feature point matching between [4] two successive pairs of frames as shown in Fig. 5. After the rectification, we perform a feature point selection in both the left and the right images and execute a feature matching. We perform the same matching for the next pair of frames, and a matching between the two successive left frames as well. This matching between successive frames yields a set of motion vectors that represents how objects moved during the last acquisition interval.

The hypothesis we utilize is that most of the feature points are bound to immobile objects in the far background such as poles, trees, houses, therefore the movement of the camera is determined by the most frequent motion vector. Any longer motion vector belongs to an object, its motion speed is modified only by this constant vector. Figure 7 shows a reconstructed depth map in order to illustrate the hypothesis. The most frequent motion vector is shown a shade of red that is detected in several background objects, anything in the foreground must have moved faster if detected. We store this vector for the remainder of the procedure.

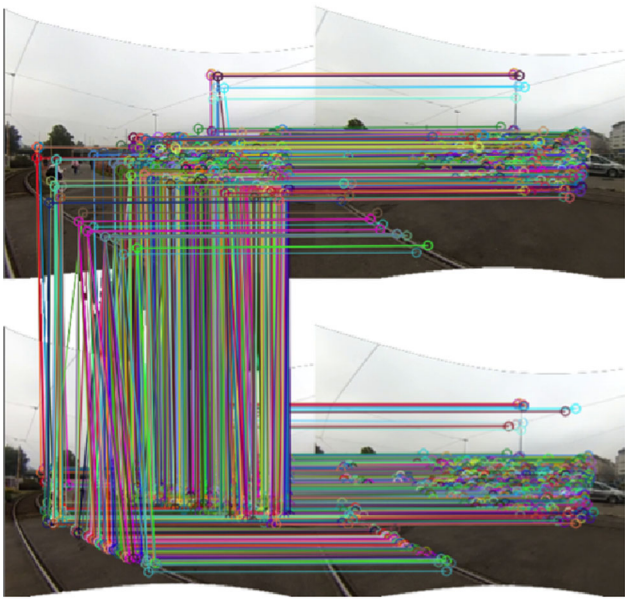

Fig. 5 Sparse feature point matching between two successive pairs of rectified frames for detecting camera movement

4 Image processing

The image processing has three main tasks: detect and classify objects that have a potential threat to the user, detect *free pathways* (no obstacles on the way), and identify stairs and crosswalks, the latter together with the traffic lamp. These all can be present in a situation at once, and there is no upper bound on the number of obstacles in a real-life scenario. In addition, no obstacles can be missed that can cause a serious damage. The processing method is very focused on general concepts allowing eventual false positive signalling.

The workflow of the image processing tasks is shown in Fig. 6. The inputs of this stage are the outputs of the preprocessing stage, and the stage outputs a set of objects with an associated threat level value, a depth map of objects that helps to find free pathways, and if there is a stair or crosswalk in close proximity in the line of sight.

4.1 Moving object detection and classification

In general, calculating the full resolution dense depth map like in [2] is the ideal approach, however computing that for each frame has high demands on the hardware for meeting the

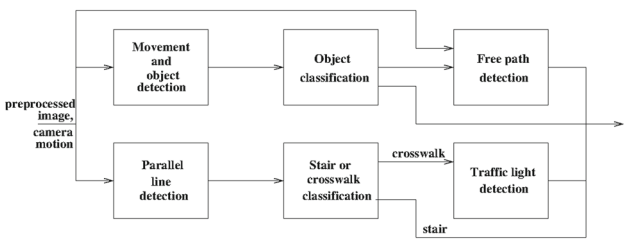

Fig. 6 The image processing workflow

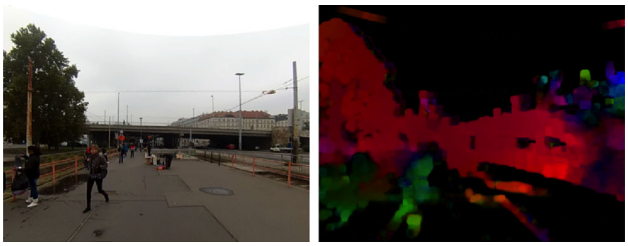

Fig. 7 Depth from motion

real time requirements. The sparse depth map computed from a small set of feature points is only reliable at those points, and the usability obtained depth image is rather questionable.

Therefore, we utilize again the motion vector results for object identification, and use dense depth maps. When the motion vectors are determined, the motion vector of the camera calculated in the stabilization procedure is subtracted from all vectors. We apply then a clustering on these motion vectors with locality and texture constraint to determine the set of objects (see Fig. 7). The locality constraint means that vectors of similar size are considered to belong to the same object only if they are in the same closure with regard to a maximum distance. The texture constraint allows neighboring points with similar texture as the feature points in the closure to be added to the detected object.

Then, we apply a labelling based on the main direction of the object obtained by fitting a line on the set of points of the object, its size, its movement speed and its position in the image. If an object has a vertical main direction, slow movement speed and it is vertically in the middle of the image, then that is labelled as a pole. If a similar object has faster movement speed and the same attributes as before, then that is labelled as a passer-by. If an object has a horizontal main direction and long motion vector, then that is labelled as vehicle. Hanging objects are searched for at the top of the image, and have slow movement speed and horizontal main direction.

Figure 8 shows an indoor scenario where we want to pass by a basket. The image on the left is the original image, where

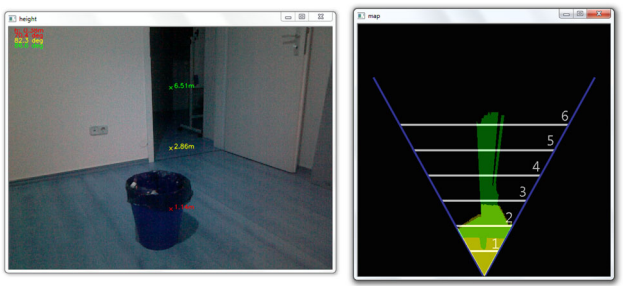

Fig. 8 Visualization of object detection and localization. The *left image* is the original image. The *right image* shows a radar-like image for the next few meters

from the middle of the bottom line we start lines radially and check the depth of points at every quarter of the full height. The right image is visualization of obstacles in the way. It shows that there is something in the way at approximately 1 m away and there is a free way straight ahead. The basket is classified as a pole, it has slow movement and a vertical main direction.

A problem of object detection, which may arise when the DPU is able to process more than 20 frames in a second, is that the error of matching and the error of depth calculated from the disparity are greater than the movement of an object in that 0.04–0.08 s. We handle this problem by matching not direct successor frames, but only every third or fifth. In that time frame, however, the scene may have undergone big changes making matching less effective.

4.2 Free path detection

The image on the right in Fig. 7 shows that no motion is detected in areas with homogeneous texture. We may also draw the consequence that there is no motion on the free path, i.e. there are no detected objects there. However, this method is not applicable in general for free path detection because it is sensitive for the movement of edges of cobblestones, holes in the asphalt and puddles that are detected as moving objects. The detection of the latter two can be considered to be good feature, however to navigate on a cobblestone road we need a more robust solution.

The assumption behind our free path detection is that the texture of the free path remains the same for the next meters. Hence, we capture first a sample from a small region in the middle in the bottom of the camera image. Then, we compute the histogram of each color channel for that region, and use a histogram backprojection for the whole image. The result is shown in Fig. 9, the free pathway is on a cobblestone road, which is difficult to handle by any other texture based technique. In the resulting image, the free passway is highlighted, but not passers, poles or any other obstacles. Moreover, this method is able to detect holes in the asphalt and puddles as those have different color compositions. This method works both indoors and outdoors.

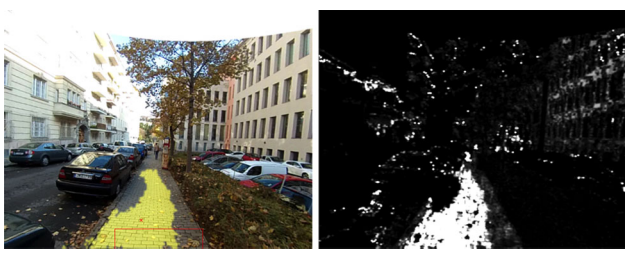

Fig. 9 Detection of free route for the next meters. The sample is taken from *rectangle* in the *bottom* of the *left image*. The *yellow color* shows the free path, the decision is made on the *right image*

The weakness of this approach is that it may loose track if there is a change of texture of the ground, for instance when the user decides to walk on the grass. Another problem is when the user does not head in the walking direction, however such movement is detected during the stabilization phase, and in such cases free path detection can be switched off temporarily.

4.3 Ground level detection

The aim of ground level detection is to eliminate the objects, mostly walls with asphalt-like texture, which are not in the plane of the ground. Ground level estimation is based on the assumption that the largest plane in the image is the ground level itself, which is true for the vast majority of the cases, there are exceptions for instance when the camera is directed on a wall.

Our ground detection algorithm is built on the so-called RANSAC principle [4]: we randomly select three non-collinear feature points on the free pathway. The three non-collinear points determine a plane, and the height of the optical center and the rotations along the y and z axes can be calculated based on the normal vector of the plane. Using these three parameters, we can determine which feature points are in the ground plane, and for each three selected points we count the number of other points found in the same plane. This procedure is repeated for a predetermined number of times, and the highest number of coplanar points is finally declared as ground level. A feature point found not to be in the ground level determines a possible obstacles, which is an input for the signaling module. An example is shown in Fig. 10.

In general, the number of feature points on the free pathway is very small, those can be found near potholes, puddles and road borders. A problem arises when the axis of the cameras is not horizontal. This can happen when the user tilts his or her head to either side. In such cases, the ground level that is the search area of the free path detection method may seem to be a slope. Therefore, we use a threshold on the angle of the camera axis and the normal vector of the plane to eliminate these situations. As a consequence, the cases when the camera sees a wall directly or the most supported plane is on a horizontal object in general are eliminated as well.

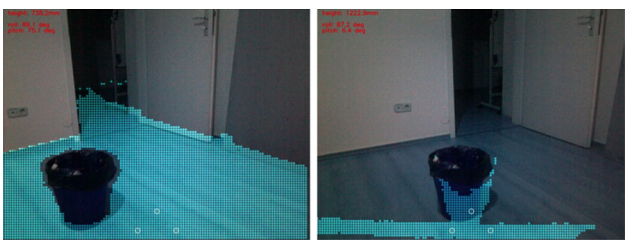

Fig. 10 Detection of the ground level. The *left image* shows a case when the three point selected at random are in the ground plane. The *right image* shows the case when a point of the three is in the ground plane

4.4 Crosswalk and stair detection

Crosswalk and stair detection are performed simultaneous tasks as these are independent from the optical flow calculations. The algorithms we developed for these two tasks have higher computation requirement, so these are not performed for every frame, these are rerun with the current image of one camera immediately when the last detection is completed.

Apart from the color properties crosswalk detection and stair detection share several structural features. When in front of a crosswalk or stairs, the two lines fitted on the left and right edges of the lanes of the crosswalk or the sides of each stair meet in a vanishing point. After applying an edge detection algorithm both stairs and crosswalk lanes appear as parallel lines that have a periodic recurrence where the distance of recurrence decreases towards the vanishing point. For distinguishing crosswalks and stairs, we can use color based properties. Our crosswalk detection procedure is trained to detect European crosswalks which are parallel lanes alternating in white and gray (asphalt color)—see Fig. 11.

The crosswalk and stair detection works as follows. We use first a flood based filtering from the edges of the picture to remove regions with more than two dominant colors. This is applicable because both stairs and crosswalks are very likely to be composed of a limited number of colors. For instance, in the case of crosswalks white and asphalt gray are dominant, while in the case of stairs there is usually only one dominant color. The second step of the method is an edge detection followed by a Hough transform for detecting parallel lines in the image. The parameters of Hough transform are set so that it only detects "long" edges. Then, we perform a hashing on the slope of the detected lines, and look for the hash bucket with the most number of elements, which is very likely to be the slope of the long edges of the white lanes of the crosswalk. In the next step, we draw a set of parallel lines perpendicular

Fig. 11 A crosswalk to be detected

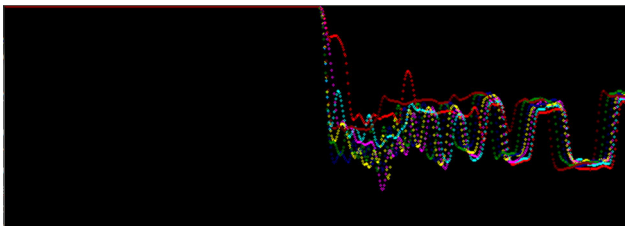

Fig. 12 Time series of lightness values along *line perpendicular* to the most frequent edge slope

to the most supported slope on the edge image, and when traversing these perpendicular lines from one edge of the image to the other, we look quasi periodical edge crossing. If found for a set of neighboring edges, there is very likely a crosswalk or a staircase, and the neighboring edges provide a bounding box.

We distinguish crosswalks and stairs based on two properties. In the case of stairs, there is only one dominant color component, while in the case of crosswalks there are two. To calculate the second property, we fit two lines to on the edges of the parallel lines in the bounding box. These two lines determine a vanishing point, if that vanishing point is an outlier compared to all other vanishing points found on the original image, then the parallel lines belong to a stair. When standing directly in front of a stair, the vanishing points of the image should be on the horizon except for one determined by the border edges of the stairs. If the stairs go downwards, then that vanishing point is an outlier below the horizon, and if the stairs go upwards, then the vanishing point is an outlier above the horizon.

To confirm a crosswalk, we use an additional filter. We create a set of time series from the lightness values along the perpendicular lines from one edge of the picture to the other as shown in Fig. 12, and compare those time series to the time series of a reference crosswalk with dynamic time warping (DTW). If the distance calculated with DTW is less than a threshold, then the perpendicular line goes through the crosswalk. Finally, we form a bounding box from neighboring parallel lines within the distance threshold, which selects the crosswalk in the image as shown in Fig. 13. With this method, we are able to find crosswalks in shades as well.

The main problems of crosswalk detection and stair detection are vehicles and passerbys who break the parallel lines. However, the coordinates of detected moving objects are available for this procedure, so it is possible to re-evaluate the region after the object has moved away.

4.5 Traffic lamp detection

Traffic lamp detection is triggered, when the crosswalk and stair detection routine returns with at least one positive hit.

Fig. 13 The bounding box of the crosswalk

Detecting traffic lamps is far from trivial, the search space is practically the whole image and the lamp itself is very small, it has the size of only a few pixels. As a blind user would be interested in traffic lamps connected to a crosswalk, we can limit the search space to the region above detected crosswalk candidates. Another input we take into account to further limit the search space are objects detected as poles, because crosswalk lamps are usually attached to them. Finally, we apply color filtering to find the lamps. In Europe, crosswalk lamps are either red or green within a black cover. The red color filter is particularly effective as it finds only traffic lamps and the red car back lamps, which are immediately dropped because of the region constraints.

5 Audio mapping of the visual world

The analysis of the traffic situation has three inputs: the set of objects moving in the user's frame of reference, the set of crosswalk or stair candidates and the set of traffic lamps. The traffic analysis module assigns a threat level to each object detected. The threat level is based on the size, distance and relative speed of the object, and whether the motion vector of the object is on a collision course with the free path of the user if the object keeps it movement speed and trajectory.

We use the superposition of audio signals to inform users about the threats. Speech generation as an alternative solution is impractical in view of quickly moving object like vehicles, where telling the user about the threat takes more time than the threat persists. The audio is produced in a way that it does not suppress environmental noises important for blind people for navigation.

We use six different sound schemes (audio icons) to describe the environment: moving object, stairs up, stairs down, traffic lamp, hole and obstructing subject. The threat level of the object is reflected in the frequency of the sound scheme, an object with a higher threat level have a higher frequency. The distance is expressed with the sound volume, the volume is inversely proportional to the distance, so that a loud sound would make the blind user stop moving. We inform the user about the relative position of the threat with the stereo balance of the sound scheme, the change of balance from one side to the other indicates that the threat crosses the current movement trajectory of the user.

Note that, frequencies and sound levels are free parameters: there is no single choice how to use them, or what they shall represent. In our tests, our configuration was found more comfortable by our test users.

6 Testing

During the development of the prototype, we used three types of cameras. For development we used Contour+ wearable cameras with 1280 × 780 resolution and 25 fps recording speed. These cameras have a close API, so in real time tests we used a pair Logitech QuickCam Pro 9000 USB cameras. As DPU, a laptop computer with Intel Core i5 3210M 2.5 GHz processor an 8 Gb RAM was used. We also tested a smartglass product with an integrated stereo camera system, a Vuzix 920AR, which however was not able to transfer both images at the same time through the USB connection to the computer.

The prototype operates in real time, in 1 s 25 image pairs are processed with the object detection algorithm, and 1–3 images with the crosswalk and stair detection procedure in average.

Our tests indicate that our audio mapping system requires at most 2 min practice to understand how it works. One must add that blind people do not like to use earphones because they rely on their senses rather than even smooth and continuous signals. That is why, we have reduced the signal generation as low as possible. In our demonstration tests, people could walk through safely in a very crowded exhibition site and even on a playground full of running children without any additional help, prior knowledge or practice.

Over 50 people have tried our solution including blind and visually impaired, children, and non-professional users in schools, playgrounds, exhibitions, and high traffic areas in Budapest. Feedbacks were very positive, nevertheless some of visually impaired people have found our solution still loud and depriving.

Figure 14 shows a picture where a volunteer tried our prototype at the Hungarian Innovation TechShow on 30 May 2014. In the demonstration, a blindfold was attached on helmet, anyone could try it. The volunteer was told the meaning of the different sound schemes, and shortly after a brief training he was able to navigate out of the demonstration hall and back, avoiding any collisions.

Fig. 14 A volunteer tries the prototype at the public demonstration

7 Conclusion

In this article, we presented our blind guide system, which is designed for helping blind and visually impaired people in their indoor and outdoor movement by providing them real time information about the environment and the traffic situation. The system is based on a stereo camera based vision system and generates audio signals for aiding the navigations. The data acquisition modules are mounted on a wearable helmet, and the processing unit fits in a backpack.

The DPU in the current prototype is limited to video inputs. Integrating a mobile data communications module and a SIM card into the DPU allows the system to connect to the services of public transportation companies. The information available through this network communication can be used to help in identifying buses, and thus help visually impaired people to get on the right bus identified by the video processing module.

Acknowledgments The development of this prototype was supported by Magyar Telekom and Gaia Software Ltd. Publishing of this work were funded by the European Union and co-financed by the European Social Fund under the Grant No. TÁMOP-4.2.2.D-15/1/KONV-2015-0006 (MedicNetwork).

References

1. ARGUS Consortium (2012–2013) Assisting personal guidance system for people with visual impairment. http://www.projectargus.eu/. Accessed 28 May 2014
2. Balakrishnan G, Sainarayanan G, Nagarajan R, Yaacob S (2007) Wearable real-time stereo vision for the visually impaired. Eng Lett 14(2):6–14
3. CASBliP (2009) Cognitive aid system for blind people. http://www.casblip.com/. Accessed 28 May 2014
4. Davies ER (2012) Computer and Machine Vision: Theory, Algorithms, Practicalities, 4th edn. Elsevier, Amsterdam (ISBN 978-0-12-386908-1)
5. Ding B, Yuan H, Jiang L, Zang X (2007) The research on blind navigation system based on RFID. In: International conference on wireless communications, networking and mobile computing (WiCom'07), pp 2058–2061
6. Gude R, Østerby M, Soltveit S (2008) Blind navigation and object recognition. Tech. rep., University of Aarhus, Denmark
7. Gustafson-Pearce O, Billett E, Cecelja F (2005) Tugs—the tactile user guidance system. Tech. rep., Brunel University, UK
8. van der Heijden F, Regtien P (2005) Wearable navigation assistance—a tool for the blind. Meas Sci Rev 5(2):53–56
9. Hub A, Diepstraten J, Ertl T (2005) Augmented indoor modeling for navigation support for the blind. In: CPSN, pp 54–62
10. Kanna V, Prasad PG, Amirtharaj S, Prabhu N, Anderson C (2011) Design of a fpga based virtual eye for the blind. In: 2nd international conference on environmental science and technology, vol 2, pp 198–202
11. Koskinen S, Virtanen A (2004) Navigation system for the visually impaired based on an information server concept. In: Mobile venue. http://virtual.vtt.fi/virtual/noppa/mvenue/navigationsystemforthevisuallyimpairedbasedonaninformation.pdf
12. Kulyukin V, Gharpure C, Sute P, De N, Nicholson GJ (2004) A robotic wayfinding system for the visually impaired. In: Proceedings of the sixteenth innovative applications of artificial intelligence conference (IAAI'04). AAAI/MIT Press, Cambridge, pp 864–869
13. Lacey G, Rodríguez-Losada D (2008) The evolution of guido. IEEE Robot Autom Mag 15(4):75–83
14. Loomis JM, Golledge RG, Klatzky RL (1998) Navigation system for the blind: auditory display modes and guidance. Presence 7(2):193–203
15. Loomis JM, Golledge RG, Klatzky RL (2001) Fundamentals of Wearable Computers and Augmented Reality, Chap. GPS-Based Navigation Systems for the Visually Impaired, 13. Lawrence Erlbaum Associates, Mahwah, pp 429–438
16. Pressl B, Wieser M (2006) A computer-based navigation system tailored to the needs of blind people. In: Miesenberger K, Klaus J, Zagler WL, Karshmer AI (eds) ICCHP 2006. LNCS, vol 4061, pp 1280–1286
17. Ran L, Helal S, Moore S (2004) Drishti: an integrated indoor/outdoor blind navigation system and service. In: Second IEEE international conference on pervasive computing and communications (PerCom'04), pp 23–30
18. Rao SK, Prasad AB, Shetty AR, Bhakthavathsalam CR, Hegde R (2012) Stereo acoustic perception based on real time video acquisition for navigational assistance. CoRR. arXiv:1208.1880