Improving Indoor Mobility of the Visually Impaired with Depth-Based Spatial Sound

Simon Blessenohl[†,‡] Cecily Morrison[†] Antonio Criminisi[†] Jamie Shotton[†]

[†]Microsoft Research [‡] University of Tübingen

Abstract

We present a novel system to help visually impaired people to move efficiently and safely in indoor environments by mapping input from a depth camera to spatially localized auditory cues. We propose a set of context-specific cues which are suitable for use in systems that provide minimal audio feedback and hence reduce masking of natural sounds compared to the audio provided by general-purpose sense substitution devices. Using simple but effective heuristics for detecting the floor and the side walls, we propose auditory cues that encode information about the distances to walls, obstacles, the orientation of the corridor or room, and openings into corridors or rooms. But the key to our system is the use of a spatial sound engine that localizes the generated sounds in 3D. We evaluate our system, comparing with [7, 16]. Our preliminary pilot study with ten blindfolded participants suggests that our system was more helpful for spotting smaller obstacles on the floor, though neither system had a significant edge in terms of walking speed or safety.

1. Introduction

The increasing ability to automatically understand the environment based on visual data will have huge implications for assistive technologies for the visually impaired. Technology is advancing rapidly on three fronts: camera sensing, automatic real-time analysis and understanding of the data, and in non-visual forms of output to the user. In this paper, we investigate the combination of a head-mounted depth camera as input, software to interpret the depth image, and a 3D sound engine to provide spatially-localized auditory cues to the user. Our goal is to improve the efficiency and safety of visually impaired users as they walk around indoor environments. In particular, we (i) address the problem of avoiding obstacles which may slow a user down even when using a cane, and (ii) aim to increase the spatial awareness of the user and overcome the problem of veering [9] by giving information about the orientation of corridors and the proximity of walls.

Camera technology is progressing on many fronts. In this work we exploit advances in depth sensing: small low-power depth cameras are now becoming available, and these greatly simplify the problem of understanding the 3D structure of the local environment. While currently of limited use outdoors, there are important indoor scenarios that can benefit today from depth sensors, and we expect that depth sensing will advance to the point where it can be applied more broadly.

Existing vision substitution technologies can be divided into two categories depending on whether they interpret the input image before generating the non-visual output. The first category aims for a *general-purpose* sense substitution, by converting the *raw* visual information into output, using *e.g.* sound [7, 15, 16]. As these systems do not interpret the visual information before generating the non-visual output, they rely on the brain's plasticity to learn the non-trivial mapping of the resulting sound patterns into a mental representation of the environment. While versatile, such general-purpose sense substitution approaches are potentially tiring on the user who is constantly bombarded with information. The second category, of which our approach is a member, instead tries to provide output cues only for *specific* patterns, such as obstacles [2, 5, 6], obstacle-free paths [12], and walls, by generating cues based on an *interpretation* of the input image. These interpretations typically abstract the complexity of the raw input image and thus help reduce sensory overload. Minimizing audio feedback also avoids masking natural sounds which is one of the key drawbacks of general-purpose sense substitution systems with audio output.

Non-visual output technology, which includes auditory and haptic [13], is also advancing. We follow many approaches in using sound, but go beyond traditional stereo in using a 3D spatial sound engine: the sounds in our system are localized in 3D space to coincide with the 3D position of

Figure 1. Our prototype. In the experiment, the laptop was carried by the experimenter walking behind the participant.

the real obstacle, wall, or corridor. We hope this might allow for a more precise and intuitive understanding of location information from sound than other mappings, especially for visually impaired users who may have superior sound localization abilities [14]. Also, spatial sound based on Head-Related Transfer Functions (HRTFs) becomes more suitable for use in assistive technology as simpler techniques for generating personalized HRTFs are developed [1, 3, 17]. To our knowledge, existing work that uses a 3D spatial sound engine [4, 8] has only used a direct mapping from uninterpreted input to output, and our approach is the first to combine depth sensing, high-level interpretation of the image, and a spatial sound engine.

2. System description

2.1. Physical setup

A time-of-flight depth camera is mounted to the front of a hard hat, pointing forwards and slightly down. The camera and headphones for audio feedback are connected to a laptop which runs the software that analyzes the depth images and generates the 3D sound scape (see Figure 1). This setup is clearly impractical for deployment (weight, ergonomics, appearance, and the blocking of ambient sounds), but it proved a useful prototype for our investigations and we believe a more appropriate form-factor would not be difficult to build using even today's technology, given suitable investment.

2.2. Preprocessing

Depth frames from the camera are downsampled by a factor of 4 in each dimension to reduce both noise and the amount of processing necessary in later steps of the pipeline. This results in an input image of 94x66 pixels, which is still ample to compute the auditory cues in our system. The value of a pixel in the downsampled frame is set to the average of the corresponding pixels in the original frame. If more than half of the corresponding pixels in the original frame have an 'invalid' depth measurement (*e.g.* the pixel is too far from the sensor), the pixel in the downsampled frame is also set to be invalid.

2.3. Detecting the floor and side walls

In order to give abstract auditory cues about indoor structure, the system heuristically classifies the points in the downsampled frame according to whether they belong to the floor, the right wall, or the left wall. To do so in a robust and fast way, we exploit the strong prior knowledge about the location and orientation of these planes relative to the camera, and make use of local normals for fast plane detection (for a performance comparison between a method based on local normals and full RANSAC, see [18]).

The local normal at a point is computed by taking the cross product of the differences between the camera space points at neighboring pixel positions. The local normal at $\vec{p}(x, y)$ (denoting the camera space point at pixel coordinate (x, y), with the origin being the top-left of the frame) is calculated using:

$$\begin{aligned}\vec{n}(\vec{p}(x,y)) = (\vec{p}(x,y+1) - \vec{p}(x,y-1)) \\ \times (\vec{p}(x-1,y) - \vec{p}(x+1,y))\end{aligned} \tag{1}$$

We check that $\vec{p}(x, y)$ in fact lies on the lines connecting $\vec{p}(x, y-1)$ and $\vec{p}(x, y+1)$, and $\vec{p}(x+1, y)$ and $\vec{p}(x-1, y)$. If not, this point is considered not to have an estimate of the local normal. This condition avoids issues at the intersection of two planes, for example.

Finding the floor. To find the floor, one iteration over the depth frame is performed and all pixel (x, y) are identified that satisfy the following two conditions:

$$\theta_{\min} \leq \theta \leq \theta_{\max} \tag{2}$$

where θ is the elevation angle of $\vec{n}(\vec{p}(x, y))$, and

$$h - \frac{h_{\mathrm{t}}}{2} \leq d \leq h + \frac{h_{\mathrm{t}}}{2} \tag{3}$$

where d is the plane-origin distance of the plane defined by the point $\vec{p}(x, y)$ and the local normal $\vec{n}(\vec{p}(x, y))$, h is a predefined estimate of the height of the camera, and h_{t} is the width of the tolerance interval.

For the first condition (2), we chose $\theta_{\min} = 65°$ and $\theta_{\max} = 165°$. It expresses that the local normal at this point is pointing up, slightly forward, or backwards. The thresholds are chosen to cover cases in which the camera is in parallel with the floor ($\theta \approx 90°$), rare cases in which it points slightly up relative to the floor plane, but parts of the floor are still visible ($\theta < 90°$), *e.g.* when the floor is tilted

downwards, and cases in which the camera is oriented down towards the floor ($\theta > 90°$).

For the second condition (3), we set $h = 1.8\text{m}$ and $h_\text{t} = 1\text{m}$. It expresses that the point lies on a plane whose distance to the camera indicates that it is the floor plane. It excludes points lying on tables, for instance, which have the same direction orientation as the floor plane, but define a plane whose plane-origin distance is smaller. Given the values of h and h_t for our system, we relied on the assumption that the camera is always on a distance of 1.3m to 2.3m to the floor. This assumption might seem unnecessarily weak, but it should be taken into account that the plane-origin distance estimate will not be accurate because small errors at local normals due to noise in the depth measurements at distant points can lead to large inaccuracy in the plane-origin distance estimate.

The camera space points satisfying these two conditions are then agglomeratively clustered into groups of points considered to lie on the same plane. Two points are considered to lie on the same plane if their corresponding normals are approximately parallel, and the difference between the points is perpendicular to the direction of their normals. The largest set of such points is taken to define the floor plane.

Finding the side walls. Firstly, all pixels are found that satisfy

$$-\frac{\theta_\text{t}}{2} \leq \theta \leq \frac{\theta_\text{t}}{2} \tag{4}$$

or

$$180° - \frac{\theta_\text{t}}{2} \leq \theta \leq 180° + \frac{\theta_\text{t}}{2} \tag{5}$$

where θ is the elevation angle of $\vec{n}(\vec{p}(x,y))$, and θ_t is the width of the tolerance interval. We chose $\theta_\text{t} = 90°$. (4) or (5) is met for all points which have local normals that are pointing roughly horizontally. The large tolerance interval allows for inaccuracy in local normal estimates and possible tilting of the head. The set of these points is clustered into planes using the same procedure as for the floor.

Among these planes, the estimate of the left wall is chosen as the largest plane satisfying

$$\phi_\text{min} \leq \phi \leq \phi_\text{max} \tag{6}$$

where ϕ is the azimuthal angle as measured in the XZ-plane, with $0°$ corresponding to the x-axis which points to the right of the camera. Additionally, if the floor is known, it is verified that

$$\begin{aligned}(\text{avg}\{\vec{p} \mid \vec{p}\,\text{lies on the candidate left wall}\,\})_x \\ < (\text{avg}\{\vec{p} \mid \vec{p}\,\text{lies on floor}\,\})_x\,.\end{aligned} \tag{7}$$

For the first condition (6), we set $\phi_\text{min} = -45°$ and $\phi_\text{max} = 20°$. This ensures that the plane normal is pointing roughly to the right, allowing more room for pointing to the back, which occurs when the camera does not point in the direction in which the wall is oriented, but is rotated towards it. The second condition (7) checks that the x-coordinate average of the points on the candidate left wall lies to the left of the average of the floor points.

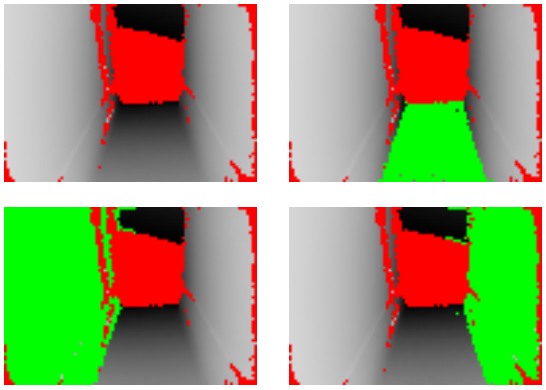

Figure 2. Result of the image plane detection. On the top left, the input frame is shown, with invalid measurements shown in red, and valid measurements ranging light gray (close) to black (far away). The remaining figures (in clockwise order) show which pixels are classified as belonging to the floor, the right wall, and the left wall. While there are some misclassifications, our approach is sufficiently robust to serve as the basis for generating auditory cues, *e.g.* about distances to walls or obstacles on the floor.

Analogous conditions are applied to find the right wall. If both a right wall and a left wall were found, it is verified that the two purported side walls are facing each other by checking that their dot product is close to 0. If that is not the case, only the side wall with more points on it is kept.

Extension step. The resulting estimates of the floor and side walls were found to reliably pick out points on the respective structures, but they often did not include all pixels which belong to them. One reason for that is that the local normals are not meaningful at the edge of such structures where the adjacent pixels belong to other planes; however, we still want to include points at these positions. To overcome this issue, a single iteration over the frame is performed which determines for each $\vec{p}(x,y)$ whether it lies on the same plane as the points currently recognized as belonging to the floor or one of the side walls. This is considered to be the case if the point lies on the same plane as randomly sampled triples of points known to be on the floor or the respective side wall. The point is then added to the set of pixels of the floor or the respective side wall. An example of the system's output of the final floor and side wall estimates is shown in Figure 2.

2.4. Generating 3D sound

The downsampled frame is passed to a set of depth-to-sound conversion routines, each of which implements a mapping from the current (and possibly previous) depth

frame to a set of sound descriptions. All sound descriptions contain the position of the sound in camera space, *i.e*. XYZ-coordinates relative to the position of the camera. A spatial sound engine generates audio on the basis of the sound descriptions. Positions of sounds in space are continuously updated during playback. For example, when the location of an obstacle relative to the user changes during playback of the sound for obstacles, the sound will move accordingly. Thereby, the sensory-motor coupling that people know from natural sounds, *e.g*. between the perceived sound and a rotating movement of their head, is simulated by our system.

Our completely unoptimized implementation runs at interactive rates (about 15 frames per second). The conversion from depth data to sound takes less than 70ms, and every other frame provided by the depth camera can be fully processed to update sound parameters. The performance of the system was measured on the laptop used in the prototype with an Intel Core i7 2.70 GHz CPU running Windows 10.

2.5. Auditory cues

Side walls. The system checks which of the walls recognized as left or right wall has the closest visible point on it, and then plays a sinusoid located at that point if it is closer than a threshold set to $1.5\mathrm{m}$ in the experiment. If the user gets closer to the side wall, the sound gets louder because the sound engine simulates it getting closer. This effect is manually enhanced in our system by scaling the amplitude of the sound. This keeps the sound very quiet at distances exceeding $1\mathrm{m}$, and quickly increases its volume if the user is in danger of walking into the side wall.

Focal area. The system identifies the closest point in a region covering approximately the central 15% of the frame. It gives an auditory cue if the closest depth in this area is less than $1\mathrm{m}$. In such cases, the user is probably walking towards a wall, or an obstacle on head height. This cue is non-abstract since it has a simple relation to the raw depth data. Consequently, it is versatile: for instance, the user can determine whether there is a wall to the right by rotating the head to the right and listening for that cue. The specific sound for this cue was chosen to be a voice repeatedly saying 'stop' until the closest point in the central area exceeds the threshold again.

Vanishing point. If the system has recognized the floor and at least one of the right or left wall, it provides a cue to indicate the orientation of the corridor or room. The cue is located in the direction of the vanishing point, which is estimated by taking the direction to which the line of intersection between the recognized wall and floor converges. As the vanishing point is located at infinite distance from the user and would hence be inaudible, the sound position was set to 5m in front of the user (in the ideal direction of the vanishing point estimate). This is intended to overcome the problem of veering. If the user keeps that sound in a direction immediately to the front of her, she would walk straight towards the end of the corridor, without being in danger of walking into side walls. A low cello note was used as sound for this cue, with a frequency one fifth below the pitch of the sound for the side wall so that simultaneous playback of the two sounds did not result in unpleasant dissonance.

Estimation of the vanishing point of the current indoor structure was used in an existing project for guidance of visually impaired people [12]. However, rather than directly providing a cue about the vanishing point, they use it in conjunction with other information to compute a suggested free walking path. Their approach for vanishing point detection relies on detecting lines in the image frame, *e.g*. at wall intersections or tiled floors. Given that our system needs to estimate the floor and side wall positions, it is computationally cheaper to estimate the vanishing point based on that information.

Openings. Cues are provided if a side wall opens up. For instance, if the corridor makes a right turn, a cue located at the end of the right wall will be generated, while at T-junctions, cues are given for both sides. To detect such openings, the system scans rows of pixels, starting at the corresponding end of the frame, *e.g*. from right to left to detect openings of the right wall. It searches for the last pixel classified as belonging to the wall, *i.e*. the end of the wall in this row. It then keeps searching for the first point which lies on the same plane as the side wall. This point does not have to be classified as part of the side wall, only as lying on the same plane as it. For instance, when the corridor makes a turn, the next point lying on the same plane as the side wall will be part of the wall which would be to the left after the turn is taken. The distance between these two points, the last on the wall and the first on the same plane, is computed and used as an estimate of the width of the opening. If the estimated width exceeds $0.5\mathrm{m}$, the opening detection test succeeded at this row. This is done for the central 15% of the rows, and if the test succeeds for more than half of them, a cue is given. Taking multiple rows into account provides robustness against noise. In the experiment, the sound for this cue was chosen to be a voice saying 'opening left' or 'opening right', located in space at the end of the side wall.

Obstacles. Small obstacles on the floor, like bins, are potentially hard to notice with non-abstract auditory cues of general-purpose sense substitution devices because they never lead to small depth values: since they are on the floor, they leave the field of view of the depth camera before they get depth values which are small compared to those at other parts of the frame, *e.g*. points at side walls. Even if the depth camera was pointing downwards with an obstacle immediately in front of the user, the distance would still be more than $1.5\mathrm{m}$ due to the height of the camera.

However, using the information of which pixels belong to the floor, such obstacles can be detected, even if they

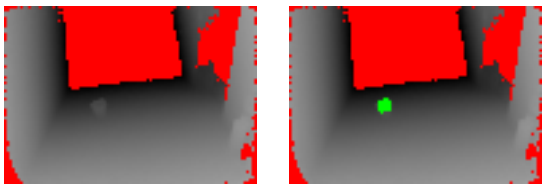

Figure 3. Output generated by the obstacle detection subsystem. On the left, the input frame is shown, with invalid measurements in red. On the right, all pixels classified as belonging to the obstacle are marked in green. The camera space point corresponding to the closest of these pixel is chosen as the position of the 3D auditory cue.

never occupy a large proportion of the frame and never lead to small depth values. The algorithm for obstacle detection firstly searches for all pixels which satisfy three conditions. Firstly, they must not be classified as part of the floor. Secondly, they must have floor pixels to the left and right of them. This means that structures attached to side walls are not considered as obstacles on the floor. Thirdly, they must have a pixel above them which is further away, *i.e*. the depth camera must be able to see a point behind them. Without this third condition, obstacle warnings would be given for structures like walls meeting in an angle greater than $180°$. The set of pixels satisfying these three conditions are considered to belong to obstacles. The pixels are then grouped into regions of adjacent pixels, and only groups of a certain minimal size are kept as representing obstacles. This reduces the number of false positives due to noise. A cue is provided at the closest obstacle pixel. Since the sound is located in space at the position of the obstacle, the user can figure out where the obstacle is and in what direction to walk to pass it. A voice repeating 'obstacle' was chosen as sound for this cue.

2.6. Stabilizing sounds

A common problem of the conversion procedures of depth frames into sounds is that they sometimes tend to produce sounds with quickly changing parameters, either due to noise or due to unfavorable surroundings (*e.g*. two obstacles at roughly the same distance). This can result in both unpleasant and confusing audio feedback. To overcome this issue, we implemented sound stabilization methods which can be used by different conversion routines. These take a set of proposed sound descriptions and return a set of descriptions of stabilized sounds, usually based on looking at the change of sound parameters through time. For example, a stabilization routine might average the position parameter of a sound over the duration of the last 500ms, or mute sounds if their position changes too quickly. In the system, such sound stabilization routines are chained, with the stabilized output of the previous stabilization routine being further stabilized according to other criteria by the following stabilization routine.

2.7. Comparison with MeloSee [7, 16]

In order to compare our system, which gives rather abstract cues, to a system aiming for general-purpose sense substitution, the MeloSee system was reimplemented [7, 16]. MeloSee uses a straightforward mapping of depth to sound: the visual field is split up evenly into a grid of 8x8 'receptive fields'. Each of the receptive fields can produce a sinusoidal sound, depending on its 'activation'. The activation of a receptive field is proportional to the average of the depth values at ten pixels within it, chosen randomly, but fixed across executions in a configuration file. This estimate of the average depth is mapped to sound intensity, with a receptive field producing no sound if the average depth in that receptive field exceeds $2.5\mathrm{m}$. Each receptive field has fixed parameters for binaural panning and sound frequency. Binaural panning depends on the horizontal position of the receptive field in the depth image and frequency on its vertical position, with receptive fields at the top of the depth frame corresponding to sinusoids of high frequency. The frequencies for the eight possible vertical positions of receptive fields are chosen to lie on a just intonation scale from C_4 to C_5.

A difference to the original implementation of the system is that due to the better range of the depth camera in our prototype, our system works at distances as close as $20\mathrm{cm}$, while their prototype was limited to a minimal distance of $50\mathrm{cm}$.

3. Preliminary mobility evaluation

3.1. Study design

We wanted to evaluate the ability of our system to help visually impaired users follow a route based on a verbal description without losing orientation or colliding with obstacles or walls. We chose to do a preliminary comparison of our system against MeloSee [7, 16], a system aiming for general-purpose sense substitution, in order to understand whether a raw, general-purpose sense substitution approach or an interpreted, specific sense substitution approach would be more helpful. Since we compared our system against a general-purpose sense substitution system in a scenario without a cane, all five sounds of our system were switched on.

Tasks. Blindfolded participants carried out two tasks in which they had to walk along routes in a real floor layout, finding possibilities to make turns and evading static obstacles. The type of task—following verbally described routes in real indoor environments—was chosen to evaluate the use of the systems to master challenges that visually impaired people might encounter. Each participant carried out both tasks using a different system for each task. Hence,

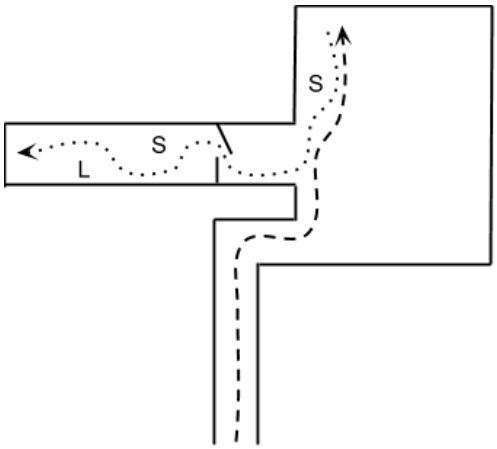

Figure 4. A sketch of the floor layout in which blindfold participants carried out the two tasks. The expected walking paths for the first and second task are shown by the dashed and dotted arrow, respectively. The positions of small and large obstacles are indicated by 'S' and 'L'.

Figure 5. The corridor in the last part of the second task. Participants had to evade a small and a large obstacle on different sides.

each participant used both our system and MeloSee, but not both systems on the same task. We do not assume the tasks to be comparable as we wanted to use realistic routes on real floor layouts with obstacles. As such, we could not assume them to be equally difficult.

The order of the two tasks was kept fixed, but the order of the systems used, and hence which system was used for which task, was randomized across participants. Participants were asked to solely rely on the audio feedback provided by the systems and not use their hands to feel where walls are. They were instructed to walk as quickly as they were comfortable with, avoiding the need of the experimenter to intervene. Interventions were made when the participant was about to walk into a wall or an obstacle, or when she lost orientation after missing the possibility to make a turn she was instructed to take.

Participants knew that there could be obstacles of various sizes. However, the number, location and size of obstacles were not known to the participants. Figure 4 and Figure 5 show the floor layout and a corridor with a small and a large obstacle.

Protocol. Before the participant was blindfolded, the audio feedback of the two systems was explained, but not demonstrated to her. Also, general usage advice for both systems was given. For instance, the importance of head movements was emphasized for the MeloSee system, as suggested in their paper [16]. Then, the two routes they had to walk were verbally described to them.

After this introduction, participants were blindfolded and the respective first system was switched on. Thus, participants never had simultaneous visual impression of the surroundings and auditory feedback. They were given a short structured introduction to the first system while they were hearing its sounds. This familiarization period lasted for about two minutes, in which they were lead through two situations: firstly, walking straight towards a wall, starting from a distance of about $3\mathrm{m}$, until being close enough to reach out and touch it, and, secondly, walking with a wall to their side while veering and coming closer to it.

They were then lead to the beginning of the first task and the task description was repeated to them. After carrying out the task, the system was swapped, they received the structured introduction for the other system, and carried out the second task using that system. Immediately after the experiment, they filled out a questionnaire.

Participants. Ten participants took part in a pilot study, aged between 18–26. They were not paid for their participation. Participants were expected to have seen the floor layout a small number of times before as it was carried out in the basement area of the building they had been working in for about six weeks. Thus, they could have potentially relied on visual memory, except for obstacle avoidance. However, as they were blindfolded before being lead to the area of the task, they did not have an immediate visual impression of the room, and generally reported that they had been completely disoriented.

Data analysis. Two measures were recorded: the time needed by a participant to walk the route, and the total number of interventions necessary. No distinction was made between the kinds of interventions (orientation lost, obstacles, walls). Thus, this measure aggregates several mobility safety aspects.

For each of our two tasks, we have data from five different participants for each of the two systems: from the total ten participants, five used MeloSee on the first task and five used our system. We evaluated whether there was a significant difference in the mean travelling time and number of interventions necessary for the two systems using an unpaired t-test (equivalent to one-way ANOVA for two groups), separately for each task. Thereby, we treat the two

	MeloSee	Our System
Mean time (Task 1)	2:51 $\pm$ 1:11	2:44 $\pm$ 1:25
Mean interventions (Task 1)	1.4 ± 1.11	1.2 ± 1.04
Mean time (Task 2)	2:56 $\pm$ 0:37	3:15 $\pm$ 1:06
Mean interventions (Task 2)	2.2 ± 2.04	0.8 ± 1.04

Table 1. Mean time needed to complete the task (in minutes) and the mean number of interventions, with 95%-confidence intervals. For each system and task, data was collected from five different participants.

tasks as two distinct between-subject experiments, not assuming the tasks to be equally difficult.

The questionnaire asked for the level of agreement to eight different comparative statements, such as "I found the sounds used in the first system less intrusive than those in the second system." on a 5-point Likert scale from "strong disagreement" to "strong agreement". Again, ANOVA was applied to test for significant differences in the mean responses given for the two systems. In addition, the questionnaire asked for general comments on the comparative advantages and disadvantages of the two systems. Responses from the ten participants were aggregated, replacing the "first" and "second" system of each participant with "MeloSee" or "our system" depending on which system that participant used first.

3.2. Results

The mean time to walk the routes and the mean number of interventions necessary are given in Table 1. No conclusions can be drawn at a significance level of 0.05 about one system allowing faster or safer performance in one of the tasks. Generally, a large intersubject variability was observed. For example, travelling times between different subjects in the first task ranged by a factor of 2 for both MeloSee and our system.

For none of the questionnaire questions, a significant deviation of the mean from 3, the midpoint on the 5-point Likert scale, was observed. Again, the responses of the subjects to the comparative questions between the two systems showed a large intersubject variability.

In particular, based on the feedback on the questionnaire, neither our system with all five sounds switched on, nor MeloSee was found to be superior to the other system in terms of intrusiveness of sounds. The mean agreement of the ten responses to the statement "I found the sounds used in our system less intrusive than those in MeloSee." (with the system names being replaced by "the first system" and "the second system") was 3.5 ± 1.03, where 3 is the neutral midpoint of the scale.

In the open questions, some users reported to have relied on the sound at the vanishing point. It was pointed out that this gave them a feeling of orientation without vision that MeloSee lacks. Explicit contextual cues about obstacles and openings were found helpful. On the other hand, the versatility of MeloSee's audio feedback was praised, although, in comparison to our system, it was pointed out that there is the danger of becoming used to a constant, fairly loud sound-level, so that dangerous situations like coming close to a wall are not easily recognized.

3.3. Discussion

All participants were able to navigate the intended routes with few safety issues regardless of system used. The results do not suggest that one system enables better mobility than the other. However, the number of participants was low and there was high intersubject variability which suggests that significance is unlikely to be achieved. It is possible that some sighted people feel anxious when walking without sight and therefore walk more slowly than others who feel more confident using other senses. This suggests that an important next step is to test the two systems on comparable routes with blind participants.

The results suggest that participants were aware of obstacles with our system and could make use of the spatial sound to get an idea of the location of the obstacle and on which side to pass it. Although no statistical significant conclusion can be drawn about the number of interventions necessary in the second task being lower with our system than with MeloSee, none of the participants using MeloSee reported to have recognized the smaller of the two obstacles, even if they passed the obstacle without the need of an intervention. Since the obstacle was a bin in a corridor, the experimental setup made it possible to evade it by chance. However, it might be possible to make MeloSee more effective for small obstacles by increasing the range in which it produces sounds to depths greater than $2.5\mathrm{m}$, so that a small obstacle stays within the audible cone of the system for a longer time, even if the participant does not look down at the floor. On the other hand, increasing the threshold of MeloSee results in smaller volume differences at closer distances and thus might affect the performance in other respects.

We had expected a difference in a sense of intrusion between the two systems, which we did not see. That our system had all sounds switched on lead to almost constant audio feedback, in particular about close walls and the vanishing point. Switching on all sounds seemed necessary since participants did not use a cane, but it meant that one of the design goals of our system, less masking of natural sounds when used in conjunction with a cane, could not come into effect.

In general, while the tasks might somewhat capture the indoor mobility challenges faced by visually impaired people, it should be kept in mind that participants in our pilot were not visually impaired, and it is expected that visu-

ally impaired people have very different skills in handling artificial sounds, just as they have very different skills in handling natural sounds [14]. Also, participants only had a short time to familiarize themselves with the systems, and long-term use might greatly alter the achieved performance, possibly increasing walking speed. For MeloSee, long-term learning over a time in which the system was not used was found to have a positive effect on the performance in a navigation task [16].

4. Conclusions

When considering assistive technology that might be adopted in the near future by more than just small groups of the visually impaired community, it seems more likely that visually impaired people are willing to use devices giving minimal auditory feedback (*i.e.* only in very specific contexts, potentially on-demand) which transmits information that they could not easily get using the white cane. An analogy can be made between 'augmented vision' for sighted people, *e.g.* using smart glasses, and 'augmented hearing' for visually impaired people, using a system like ours. Both types of technology make one sense more powerful by artificially inducing sensory perceptions on this sense to provide information which would normally not be accessible to it. Just like sighted people want artificial visual information to interfere as little as possible with the relevant natural visual information provided by the environment, visually impaired people might prefer artificial auditory information that interferes as little as possible with relevant natural acoustic information.

In this paper, we suggested auditory cues which are suitable for such intelligent mobility aids that minimize interference. Also, we combined such abstract cues with spatial sound to give location information, e.g. about obstacles, in an intuitive way. Results from a pilot experiment indicate that such specific cues would be useful to visually impaired people, possibly as useful in an indoor mobility setting as sounds of general-purpose sensory substitution devices, which have the disadvantage of being less suitable for use in systems that minimize audio feedback.

There are many possible extensions of our system. Additional spatially localized cues could be provided, *e.g.* for faces. Going a step further, face recognition would allow the system to inform the visually impaired user about who is approaching her. Generally, object recognition techniques are potentially useful in the area of assistive technology for the visually impaired. In the specific scenario we have been investigating, a classifier for obstacles (such as [2]) could be integrated to inform the user about the type of obstacle in front of her. Furthermore, it would be desirable to introduce an interactive component to the system so that the user can specifically require certain cues, e.g. about the orientation of the room. Careful sound design, probably replacing spoken voice by iconic sounds, has the potential of making the system more pleasant to use, decreasing the cognitive load on the user [11] and supporting the localization of structures with spatial sound.

In terms of evaluation, only one preliminary pilot has been conducted so far. Besides the obvious need to run a larger study with visually impaired participants, different audio output setups could be compared. Use of bone-conducting headphones and restricting audio feedback to one ear (see [14] for evidence that visually impaired people are good at localizing spatial sound monaurally) are just two ways in which future systems could potentially reduce interference with natural sounds.

References

[1] J. Ahrens, M. R. P. Thomas, and I. Tashev. HRTF magnitude modeling using a non-regularized least-squares fit of spherical harmonics coefficients on incomplete data. In *APSIPA Annual Summit and Conference*, 2012. 2

[2] A. Bhowmick, S. Prakash, R. Bhagat, V. Prasad, and S. M. Hazarika. IntelliNavi: Navigation for blind based on Kinect and machine learning. In *Multi-disciplinary Trends in Artificial Intelligence*, pages 172–183. 2014. 1, 8

[3] P. Bilinski, J. Ahrens, M. R. P. Thomas, I. J. Tashev, and J. C. Platt. HRTF magnitude synthesis via sparse representation of anthropometric features. In *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 4468–4472, 2014. 2

[4] G. Bologna, B. Deville, T. Pun, and M. Vinckenbosch. Transforming 3D coloured pixels into musical instrument notes for vision substitution applications. *EURASIP Journal on Image and Video Processing*, vol. 2007, 2007. 2

[5] M. Brock and P. O. Kristensson. Supporting blind navigation using depth sensing and sonification. In *Adjunct Proceedings of the 2013 ACM Conference on Pervasive and Ubiquitous Computing*, UbiComp 2013, pages 255–258, 2013. 1

[6] V. Filipe, F. Fernandes, H. Fernandes, A. Sousa, H. Paredes, and J. Barroso. Blind navigation support system based on Microsoft Kinect. *Procedia Computer Science*, 14:94–101, 2012. 1

[7] V. Fristot, J. Boucheteil, L. Granjon, D. Pellerin, and D. Alleysson. Depth-melody substitution. In *20th European Signal Processing Conference (EUSIPCO-2012)*, pages 1990–1994, 2012. 1, 5

[8] J. González-Mora, A. Rodríguez-Hernández, L. Rodríguez-Ramos, L. Díaz-Saco, and N. Sosa. Development of a new space perception system for blind people, based on the creation of a virtual acoustic space. In J. Mira and J. V. Sánchez-Andrés, editors, *Engineering Applications of Bio-Inspired Artificial Neural Networks*, volume 1607 of *Lecture Notes in Computer Science*, pages 321–330. 1999. 2

[9] D. Guth and R. LaDuke. The veering tendency of blind pedestrians: An analysis of the problem and literature review. *Journal of Visual Impairment and Blindness*, 88:391–391, 1994. 1

[10] S. L. Hicks, I. Wilson, L. Muhammed, J. Worsfold, S. M. Downes, and C. Kennard. A depth-based head-mounted visual display to aid navigation in partially sighted individuals. *PLoS ONE*, 8(7), 2013.

[11] R. L. Klatzky, J. R. Marston, N. A. Giudice, R. G. Golledge, and J. M. Loomis. Cognitive load of navigating without vision when guided by virtual sound versus spatial language. *Journal of Experimental Psychology: Applied*, 12(4):223–232, 2006. 8

[12] A. Landa-Hernández, H. Casarubias-Vargas, and E. Bayro-Corrochano. Geometric fuzzy techniques for guidance of visually impaired people. *Applied Bionics and Biomechanics*, 10(4):139–157, 2013. 1, 4

[13] Y. H. Lee and G. Medioni. Wearable RGBD indoor navigation system for the blind. In L. Agapito, M. M. Bronstein, and C. Rother, editors, *Computer Vision - ECCV 2014 Workshops*, volume 8927 of *Lecture Notes in Computer Science*, pages 493–508. 2015. 1

[14] N. Lessard, M. Pare, F. Lepore, and M. Lassonde. Early-blind human subjects localize sound sources better than sighted subjects. *Nature*, 395(6699):278–280, 1998. 2, 8

[15] P. B. Meijer. An experimental system for auditory image representations. *IEEE Transactions on Biomedical Engineering*, 39(2):112–121, 1992. 1

[16] C. Stoll, R. Palluel-Germain, V. Fristot, D. Pellerin, D. Alleysson, and C. Graff. Navigating from a depth image converted into sound. *Applied Bionics and Biomechanics*, vol. 2015, 2015. 1, 5, 6, 8

[17] I. Tashev. HRTF phase synthesis via sparse representation of anthropometric features. In *Information Theory and Applications Workshop*, 2014. 2

[18] H. W. Yoo, W. H. Kim, J. W. Park, W. H. Lee, and M. J. Chung. Real-time plane detection based on depth map from Kinect. In *44th International Symposium on Robotics*, pages 1–4, 2013. 2