Depth and motion cues with phosphene patterns for prosthetic vision

Alejandro Perez-Yus Jesus Bermudez-Cameo Gonzalo Lopez-Nicolas
Jose J. Guerrero
Instituto de Investigacion en Ingenieria de Aragon (I3A), Universidad de Zaragoza, Spain
{alperez, bermudez, gonlopez, josechu.guerrero} @unizar.es

Abstract

Recent research demonstrates that visual prostheses are able to provide visual perception to people with some kind of blindness. In visual prostheses, image information from the scene is transformed to a phosphene pattern to be sent to the implant. This is a complex problem where the main challenge is the very limited spatial and intensity resolution. Moreover, depth perception, which is relevant to perform agile navigation, is lost and codifying the semantic information to phosphene patterns remains an open problem. In this work, we consider the framework of perception for navigation where aspects such as obstacle avoidance are critical. We propose using a head-mounted RGB-D camera to detect free-space, obstacles and scene direction in front of the user. The main contribution is a new approach to represent depth information and provide motion cues by using particular phosphene patterns. The effectiveness of this approach is tested in simulation with real data from indoor environments.

1. Introduction

The ability to navigate and move around complex or unfamiliar environments is essential for people, and this is a non-trivial task to be automated. People solve these tasks primarily through vision, combined with their ability to memorize and learn. These tasks are even more critical for visually impaired people, since additional personal safety issues appear. While mobility aids such as the white cane are helpful in short-range navigation, the usage of cameras enable the recovery of mid- and long-range information from the environment and thus, a more effective navigation. A key issue in Navigation Assistance for Visually Impaired (NAVI) is obstacle avoidance. Different approaches for NAVI have been developed based on vision sensors such as in [47, 36, 38, 2], or with other types of sensors [12, 35, 18]. In the context of prosthetic vision, different visual processing techniques were proposed for obstacle avoidance [40, 46, 31, 29].

In the following sections we provide some background on the topic of prosthetic vision. Then, we describe the main aspects of phosphene mapping techniques and how they are usually tested with users. Finally, we describe the problem of depth and motion perception considered and the proposed contributions.

1.1. Background on prosthetic vision

Since 1968, different research works have found that electrical stimulation of the visual cortex or other parts of the visual pathway (such as retina) caused patients to perceive bright dots of light called phosphenes [6]. Thus, visual prostheses generally consist of retinal or cortical implants that apply electrical stimulation using an electrode array to generate a grid of phosphenes similar to a low resolution dot image [11].

The typical components of this technology are as follows: A small camera mounted on the eyeglasses is used for image acquisition. The images are then processed by a portable computer to convert the image data into an electronic coded signal. This signal is transferred to the implant via wireless communication and the signal finally reaches the microelectrode array causing the grid of phosphenes. Experimental results demonstrate that patients with this kind of devices can detect phosphenes at individual electrodes and they were able to develop coordination in using their visual prosthetic device [1].

1.2. Models of phospene patterns

Unfortunately, the resolution of the phosphene grid produced by visual prostheses is constrained by biology, technology and safety [30, 34], and current devices provide a few dozens of phosphenes. Therefore, implanted visual prostheses provide bionic vision with very limited spatial and intensity resolution when compared against healthy vision. According to [7] a pattern of 25×25 phosphenes allows to recognize text in a reading speed of 100 words per minute for stationary text and 170 words per minute for text moving automatically. Other related works also study performance in the task of reading [17], or finding text [13].

However, other tasks like face recognition require hundreds of phosphenes [41, 45].

Given the highly limited resolution, important efforts have been performed on the application of vision algorithms to improve the phosphene patterns for prosthetic vision [4]. For example, vision processing can make better use of the limited resolution by highlighting salient features such as edges [30, 32, 14]. Currently, the way to process and code the image information to the low resolution device to be useful and meaningful is still an open issue.

Moreover, traditional works generally assume regular phosphene patterns to be created with the prosthetic vision device. However, there is clinical and biological proofs that phosphene patterns are irregular and patient-specific [39], [28]. Still, regular patterns are usually assumed, and works that cope with irregular phosphene maps generally consider close to regular patterns where small spatial shifts in phosphene locations and electrode dropouts are modelled [42], or only irregular phosphene shapes are considered over a regular grid [25]. Regarding the particular shape of the visual phosphenes, there is a large variety of profiles described in the literature. For simplicity most works choose either perfectly circular or square shaped phosphenes for their simulation studies [8]. However, neither perfectly circular nor square phosphenes capture the exact shape of real phosphenes.

1.3. Simulated Prosthetic Vision

In order to avoid complex and costly trials on patients, a non-invasive method to evaluate the efficacy of visual prostheses is by means of Simulated Prosthetic Vision (SPV) [30]. Most SPV systems make use of a head mounted display with a forward facing camera, which allows fast testing of a great variety of methods while constraining the user to a particular model of bionic vision such as visual angle or resolution. A discussion about different SPV is provided in [8, 9].

Most of the current approaches used in prosthetic vision and SPV are based on basic image processing techniques [3, 4]. However, this vision-based configuration allows exploring more advanced computer vision techniques to enhance the semantics and the relevance of the information displayed to the patient [5]. For example, visual recognition can be used for enhancing the saliency of meaningful objects [22, 24]; face detection can be used for human interaction [30]; or clarity of symbolic information can be improved with image segmentation techniques [22]. Recently, [23, 43, 48] used virtual-reality-based environments to evaluate the user response with different models of visual representation.

1.4. Problem definition and contributions

As previously said, in prosthetic vision a visual scene is composed of relatively large and isolated spots of light called phosphenes. However, very low resolution images are frequently meaningless to the user. Moreover, representing depth in phosphene maps is very relevant to achieve adequate navigation, but its implementation is particularly challenging. Notice that depth perception cannot be transmitted using stereo displays because of intrinsic technical limitations of prosthetic vision. Thus, it requires alternative strategies to transmit depth such as using an iconic representation.

Our proposal consists of a perception system module (Section 3) and the iconic representation module (Section 4) for the camera-computer configuration in prosthetic vision. The goal of our perception module is to retrieve:

- The relative movement of the user in the scene.
- The orientation of the scene.
- A collision-free walkable path.

The type of camera we choose for information acquisition is an RGB-D camera, carried by the user mounted in the head. The RGB-D cameras provide a colour image with also the depth of each pixel in the image. These devices are currently in development and are subject to intensive research. In our framework, this type of information is particularly useful to reliably detect obstacles and, for example, warn of other potentially dangerous situations such as the presence of curbs or stairs [37], or detect the location of an empty chair [44]. Usually, it is assumed that man-made environments are essentially composed of three main directions orthogonal to each other. Taking this assumption into account, denoted as Manhattan world assumption, some works have been proposed for recovering the scene layout [26, 20, 16], or just the orientation of the scene [10] as we do in this work.

In the proposed iconic representation module we code the information perceived to the phosphene map. This task is challenging, since our approach tries to accommodate to the current state of technology of prosthetic visual devices. Despite the recent progress in the field, the resolution and dynamic range are still low. Moreover, related works focus on 2D information neglecting the three-dimensional nature of the world, and depth perception is lost to the user. Systems displaying depth and contrast edges in a phosphene-based display are described in [27, 30] and more recently in [33]. In [21], a semantic labelling of the image provides a representation for obstacle avoidance. Here we aim to the ambitious goal of providing depth information by designing appropriate processing algorithms to be used on the vision-based input information.

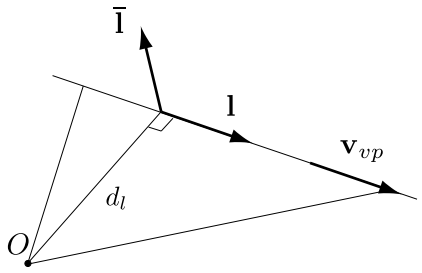

Figure 1: Components of a Plücker description of a 3D line. The direction vector $\mathbf{l}$ and the moment vector $\bar{\mathbf{l}}$. Under Manhattan World assumption a main direction $\mathbf{v}_{vp}$ is coincident with the direction $\mathbf{l}_i$ of lines following this direction and orthogonal with their moment vectors $\bar{\mathbf{l}}_i$.

In this paper, we present a novel phosphene map coding for navigation tasks based on a ground representation of the obstacle-free space as a polygon and a ceiling representation based on vanishing lines pointing towards a previously determined moving direction. The ground polygon is codified with a chess pattern to provide the effect of displacement over the ground with the relative pose obtained with the odometry. The effectiveness of the proposed representation is illustrated with real data from indoor scenes in Section 5.

2. Geometry and notation details

Consider a set of points planes and lines in a given reference. We denote $\mathbf{X} \in \mathbb{P}^3$ a 3D point in homogeneous coordinates. We denote $\mathbf{U} = \left(\mathbf{u}^\mathsf{T}, u_0\right)^\mathsf{T}$ a plane in homogeneous coordinates. We denote $\mathbf{L} \in \mathbb{P}^5$ a 3D line in Plücker coordinates composed by two vectors $\mathbf{L} = \left(\mathbf{l}^\mathsf{T}, \bar{\mathbf{l}}^\mathsf{T}\right)^\mathsf{T}$ being $\mathbf{l} \in \mathbb{R}^3$ a vector describing the direction of the line, $\bar{\mathbf{l}} \in \mathbb{R}^3$ is a vector representing the normal to a plane passing through the 3D line and the origin of the reference system O, and the ratio between its norms $d_l = \frac{\|\bar{\mathbf{l}}\|}{\|\mathbf{l}\|}$ is the minimum distance from the line to the origin of the reference system (see Fig. 1). To allow $\mathbf{L}$ being a 3D line $\mathbf{l}^\mathsf{T}\bar{\mathbf{l}} = 0$. Rays are also codified as Plücker coordinates but denoted with $\boldsymbol{\Xi} = \left(\boldsymbol{\xi}^\mathsf{T}, \bar{\boldsymbol{\xi}}^\mathsf{T}\right)^\mathsf{T}$.

Consider a reference system composed of a rotation $\mathsf{R} \in SO(3)$ and a translation $\mathbf{t} \in \mathbb{R}^3$. A change of reference of points is performed by using the linear transformation $\mathsf{T} \in SE(3)$ such that $\mathsf{T} = \begin{pmatrix} \mathsf{R} & \mathbf{t} \\ \mathbf{0}^\mathsf{T} & 1 \end{pmatrix}$. A change of reference of a plane is done through $\mathsf{T}^{-\mathsf{T}}$. Finally, a change of reference of a line or a ray is described by the linear transformation $\mathsf{G} = \begin{pmatrix} \mathsf{R} & 0 \\ [\mathbf{t}]_\times \mathsf{R} & \mathsf{R} \end{pmatrix}$.

3. Perception of free moving space and scene orientation

The proposed system includes an RGB-D camera for the perception part. In this section we describe the main subtasks as defined in the introduction: to obtain the relative movement of the user in the scene (Section 3.1), to get the orientation of the scene (Section 3.2), and to retrieve a layout of free moving space (Section 3.3).

3.1. Relative movement of the user in the scene

In robotics, the estimation of the position of the robot with respect to the starting location is called *odometry*. When the information to compute the odometry comes from a camera, it is called visual odometry. This is a classic topic in computer vision, which recently has been enhanced with the advent of RGB-D cameras. We use the algorithm from [19], which is a method for dense visual odometry estimation performed by minimizing photometric (in the RGB image) and geometric (in the inverse depth map) errors, and therefore takes advantage of the RGB-D camera.

With this method, for each frame we compute the pose $\mathsf{T}_{0,k} \in SE(3)$ that transforms the reference frame from k to 0, being 0 the initial reference frame. These transformation $\mathsf{T}_{0,k}$ consists of a rotation matrix $\mathsf{R}_{0,k} \in SO(3)$ and a translation vector $\mathbf{t}_{0,k}$. These transformation is necessary to provide sense of movement in the environment.

3.2. Orientation of the scene

In our work we assume scenes satisfy the Manhattan World assumption [10], meaning the world is organized according to three orthogonal directions, we call *Manhattan directions* or *main directions*. In order to get these directions, we perform a vanishing point extraction, since all lines directed in one of the Manhattan directions intersect in one of the three main vanishing points.

First, from the distribution of the normals of the point-cloud we obtain a set of rough candidates for being the main three directions. Then, lines are extracted from the RGB-image and clustered in main directions following a Random Sample Consensus (RANSAC) approach [15]. Assuming Manhattan directions, we can assemble the direction vectors to create the rotation matrix $\mathsf{R}_{k,Abs} \in SO(3)$, being Abs the reference of the system with the axis aligned with the Manhattan directions, called *absolute reference*. To enforce the obtained directions to be orthogonal we optimize $\boldsymbol{\omega}_{k,Abs} \in \mathfrak{so}(3)$ such that $\mathsf{R}_{k,Abs} = \exp\left([\boldsymbol{\omega}_{k,Abs}]_\times\right)$. The distance of the minimization $d_{opt} = \mathbf{v}_{vp}^\mathsf{T}\bar{\mathbf{l}}_i$ exploits the constraint that the direction vector of a 3D line $\mathbf{L}_i$ must be orthogonal to its corresponding projection plane (see Fig. 1). For considering that the original clusters could contain some misclassified lines we use a $L1$-norm as loss function.

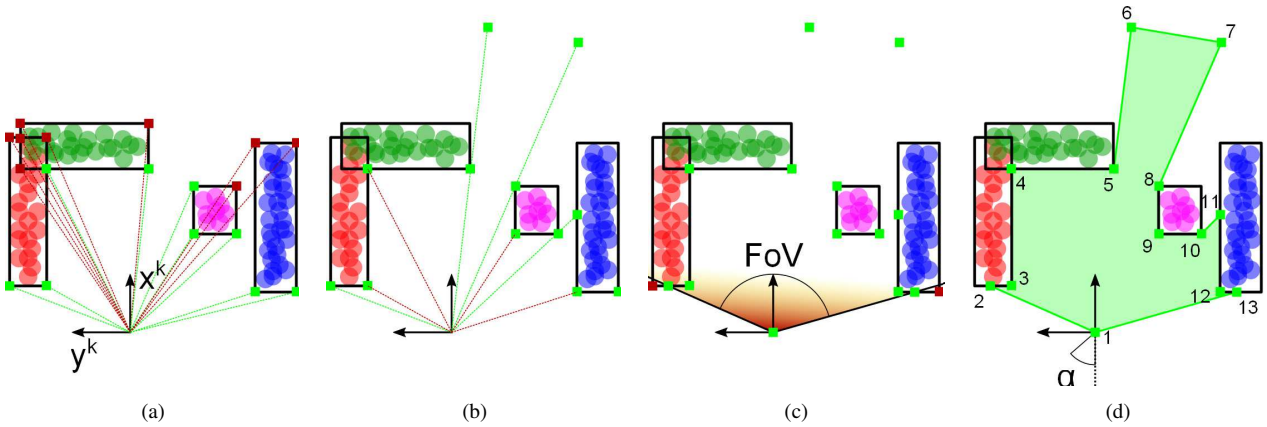

Figure 2: Four basic steps for the construction of the floor polygon following the explanation from Section 3.3. Four obstacles are drawn with the projection to the floor in different colors and bounding boxes. Valid vertices are colored in green while invalid vertices are dark red. In (d) the floor polygon is drawn in green.

Finally, the result is fine-tuned with a $L2$-norm using only a selected collection of well conditioned lines.

In practice, this procedure can be performed only once and then carried over by the odometry. For example, let us consider we obtain $R_{Abs,0}$ at first frame. At frame k we can compute the pose $T_{Abs,k}$ with $t_{Abs,k} = t_{0,k}$ and $R_{Abs,k} = R_{Abs,0} \cdot R_{0,k}$. We choose the axis in Abs to be as follows: z_{Abs} pointing upwards (to where the ceiling should be), x_{Abs} to the front of the user in that moment and y_{abs} to its left.

3.3. Perception of free space

The free space around the user is retrieved using the information from the depth camera, specifically the point cloud data. A point cloud is a set of 3D points $\mathbf{X}_i = (x_i, y_i, z_i, 1)^\mathsf{T}$, each one corresponding to a pixel in the depth camera. To speed up the algorithm, instead of making operations to the whole cloud we perform downsampling via voxel grid filter. We particularly apply a voxel size of 0.10 meters, with could reduce the cloud approximately 100 times without major loss of relevant data for this task. The point cloud can be transformed to the absolute reference frame by $\mathbf{X}^{Abs} = \mathsf{T}_{Abs,k} \cdot \mathbf{X}^k$.

Once we have our data pre-processed and in the absolute reference frame, we compute the floor plane. To do so, we have a tentative orientation of the normal of the floor plane, since it should align with the main direction corresponding to z_{Abs}. Thus, we proceed by computing the normals of the points $\mathbf{n_X}$ via principal component analysis, and selecting the points whose normal is near z_{Abs}. Then, a RANSAC procedure for planes is applied to that subset of points, and among the resulting plane candidates computes their distance to the origin u_0 and chooses as solution that with highest value of u_0. Note that we are assuming that the floor is visible and that there is no other horizontal plane below it. Again, like with the main directions, the floor plane does not need to be retrieved every frame, and it can be carried over by the odometry. This has an important advantage, since then the floor does not need to be in the image all the time and the user can be looking elsewhere.

The points of the cloud that are not classified as floor points are considered obstacles unless they are considerably higher than the person (e.g. ceiling). Since the camera is designed to be on the head, we choose $z = 0.5$ meters over the head as a safe maximum threshold to consider points out of reach. The points are then grouped in planes and clusters combining a RANSAC approach with Euclidean cluster extraction (i.e. points that are close to each other under certain threshold are grouped in clusters). Each plane or cluster is considered *obstacle*, meaning that no further semantic reasoning has been performed. To determine the layout of free space we project the points to the floor plane to reason in 2D. The layout of free space will be the polygon on the floor plane whose edges are given by the bounding boxes of the obstacles and the rays from the camera. The procedure to build the floor polygon goes as follows (see Fig. 2 for graphical explanation):

(a) We look for visible vertices in the corners of the bounding boxes of the floor projections of the obstacles. Visibility is checked by the intersection of the segments from the origin to the vertices and the segments of the bounding boxes. Also, intersection points between bounding boxes are considered.

(b) We project rays from the origin to the previous vertices to verify if they intersect with their bounding boxes or not. Those who do not intersect are extended until intersection with other bounding box or to a maximum

distance (e.g. 10 meters).

(c) We remove those vertices outside the field of view (FoV) of the camera, and include the origin as vertex.

(d) We sort the vertices clockwise with the angle α, sorting accordingly vertices with same angle (i.e. coming from (b)).

The output of the proposed procedure is a 2D polygon in the floor plane that we can transform in 3D since we know the plane equation (see Fig. 2 (d)).

4. Iconic representation of layouts

Even when the environment is known, the lack of dynamic range and resolution in prosthetic visual devices complicates the perception of depth. On the one hand, the quantification given by the low resolution hinders the possibility of stereographic vision. In the other hand, the low dynamic range dilutes the texture of landmarks that humans use for locating themselves.

In order to tackle with these perception problems, we consider using an iconic representation of the scene capable of giving support for navigation tasks. Our proposal consists of: a layout representation of the ground describing the free space and a simplified representation of the ceiling suggesting the motion direction which has been planned in a higher level. The displacement with respect the ground is evoked by using a chess pattern in the ground map. This representation, inspired by old low-resolution 3D games, is useful as an iconic description of perspective projection.

As introduced in Section 3, we estimate the free ground and represent it with a polygon. This polygon is defined in a global reference we have obtained from the main directions of the scene and the odometry we get from a RGB-D SLAM system [19]. For representing the ground, we first estimate which rays intersect with the polygon defining the layout. For this we use the Plücker polygon-ray intersection approach which works with convex polygons. Since the polygon is in general non-convex we estimate the rays intersecting the convex-hull of the polygon and then we remove the outlier points.

4.1. Plucker polygon-ray intersection method

Given a line $\mathbf{L} = \left(\mathbf{l}^\mathsf{T}, \bar{\mathbf{l}}^\mathsf{T}\right)^\mathsf{T}$ and a ray $\mathbf{\Xi} = \left(\boldsymbol{\xi}^\mathsf{T}, \bar{\boldsymbol{\xi}}^\mathsf{T}\right)^\mathsf{T}$ represented in Plücker coordinates, the side operator

$$\text{side}\left(\mathbf{L}, \mathbf{\Xi}\right) = \mathbf{l}^\mathsf{T}\bar{\boldsymbol{\xi}} + \bar{\mathbf{l}}^\mathsf{T}\boldsymbol{\xi} \tag{1}$$

returns a signed distance. The sign of this side distance defines the relative wise between the lines (see Fig. 3). If we define the sides of a convex polygon with their corresponding Plücker coordinates and we follow a clockwise sense in this definition we can determine if a ray intersects the interior of the polygon or not by using the following rule.

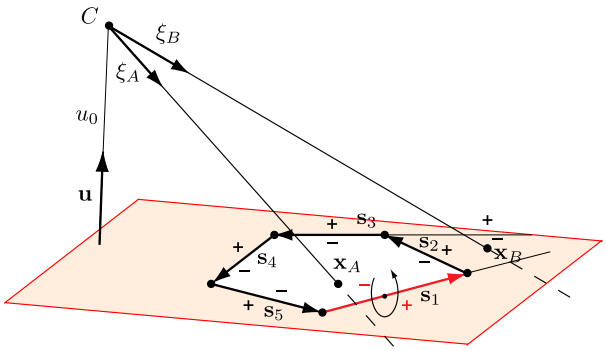

Figure 3: The sign of the distances from ray A $\left(\boldsymbol{\xi}_A, \bar{\boldsymbol{\xi}}_A\right)$ to the sides $(\mathbf{s}_i, \bar{\mathbf{s}}_i)$ are $(-,-,-,-,-)$. By contrast, the sign of the distances from ray B $\left(\boldsymbol{\xi}_B, \bar{\boldsymbol{\xi}}_B\right)$ to the sides $(\mathbf{s}_i, \bar{\mathbf{s}}_i)$ are $(-,+,-,-,-)$.

(a) Estimate the sign of the side between the ray and each side.

(b) If all the side distances have the same sign the ray intersects the interior of the polygon.

(c) If this sign is positive we are looking to the front of the polygon, if negative we are looking to the back.

Once we know the rays intersecting the convex-hull of the polygon we compute the corresponding projected points. Then, we collect the projected points which are inside the polygon and mark them with the chess pattern. This texture is parametrically defined by quantizing X and Y coordinates. Since the points are computed in the global reference (which is aligned with the vanishing points) the chess pattern follows the main directions of the scene.

5. Experiments and discussion

We have tested our method performing real world situations. Our experimental setup consisted of a head mounted RGB-D camera that was attached to a helmet. The RGB-D camera model is an Asus Xtion Pro Live, a widely used device in computer vision and robotics research. We used a laptop to record the sequences, but no direct feedback to the user (e.g. via virtual reality glasses) was tested yet at this early stage. However, this approach let us try different ways of encoding the data in phosphenic representation and analyze the problems that may emerge in a real scenario. Here we provide a qualitative insight about the method, its limitations and possibilities.

5.1. Evaluation of free space perception

First, it is important to have a reliable perception module to create a safe and useful assistive aid. The recovery of the

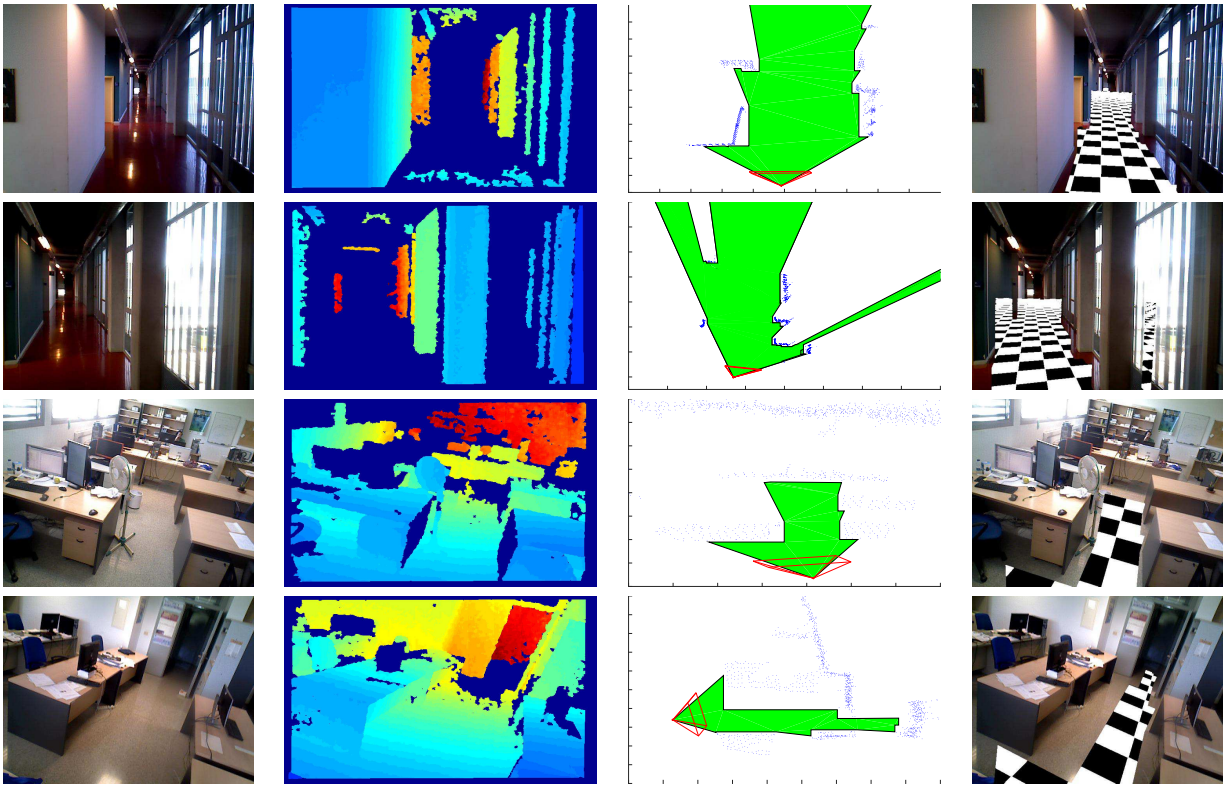

Figure 4: Each row shows an example of the perception of free moving space. The first two columns show the RGB and depth image, which has been scaled so warmer colors mean farther distances. Column 3 shows the floor plan view of the moving polygon and in column 4 the chess pattern has been overlaid to the RGB image for visualization.

main orientation of the scene works well considering the Manhattan World assumption holds for a vast majority of indoor environments. On the other hand, the visual odometry has a slight drift that is more noticeable the longer the algorithm is running. However, by re-computing the vanishing points and floor plane from time to time the effect of the drift should be minimized.

The most important part about the perception module is the obstacle detection, since this is what warns the user when he is prone to have an accident. Its proper functioning depends mostly on the sensor and its limitations. Conventional RGB-D cameras are well known for not working well at direct sunlight, which should not be problematic when staying indoors. In this situation most obstacles are detectable, with the exception of certain materials that absorb or reflect the infrared light emitted by the camera. In our experiments, the only type of surface that remains always undetected was glass, which is also undetectable with conventional cameras.

We show some examples in Fig. 4, including RGB and Depth captures, along with a 2D floor projection of the scene, where the obstacles are drawn as blue points and the free moving space is the green polygon; and the overlay of the corresponding chess pattern to the color image. The first two rows correspond to a corridor scenario: a successful case and an unsuccessful one. The former not only is able to recover the main path to follow the corridor, but it also shows the beginning of new paths at the left, that the user may like to know to explore the environment. The second example shows the opposite: a misdetection on the left wall and the glass surface at the right, showing some misleading free space, which could be problematic. The third and fourth row in Fig. 4 show another environment, in this case an office place full of tables and other obstacles. In both cases the algorithm is able to show the main path to follow and some additional sidetracks. The last row finds a traversable path through the door. However, it leaves some actual free space undetected at the left, next to the door.

5.2. Evaluation of hosphenic representation

Regarding the phosphenic representation, we show some examples in Fig. 5. Our iconic representation of the floor includes three levels of intensity of the phosphenes (black, gray and white), to color the chess pattern in white and gray. Turned off phosphenes (black) mean no walkable path. In case only one level of intensity could be used, an alterna-

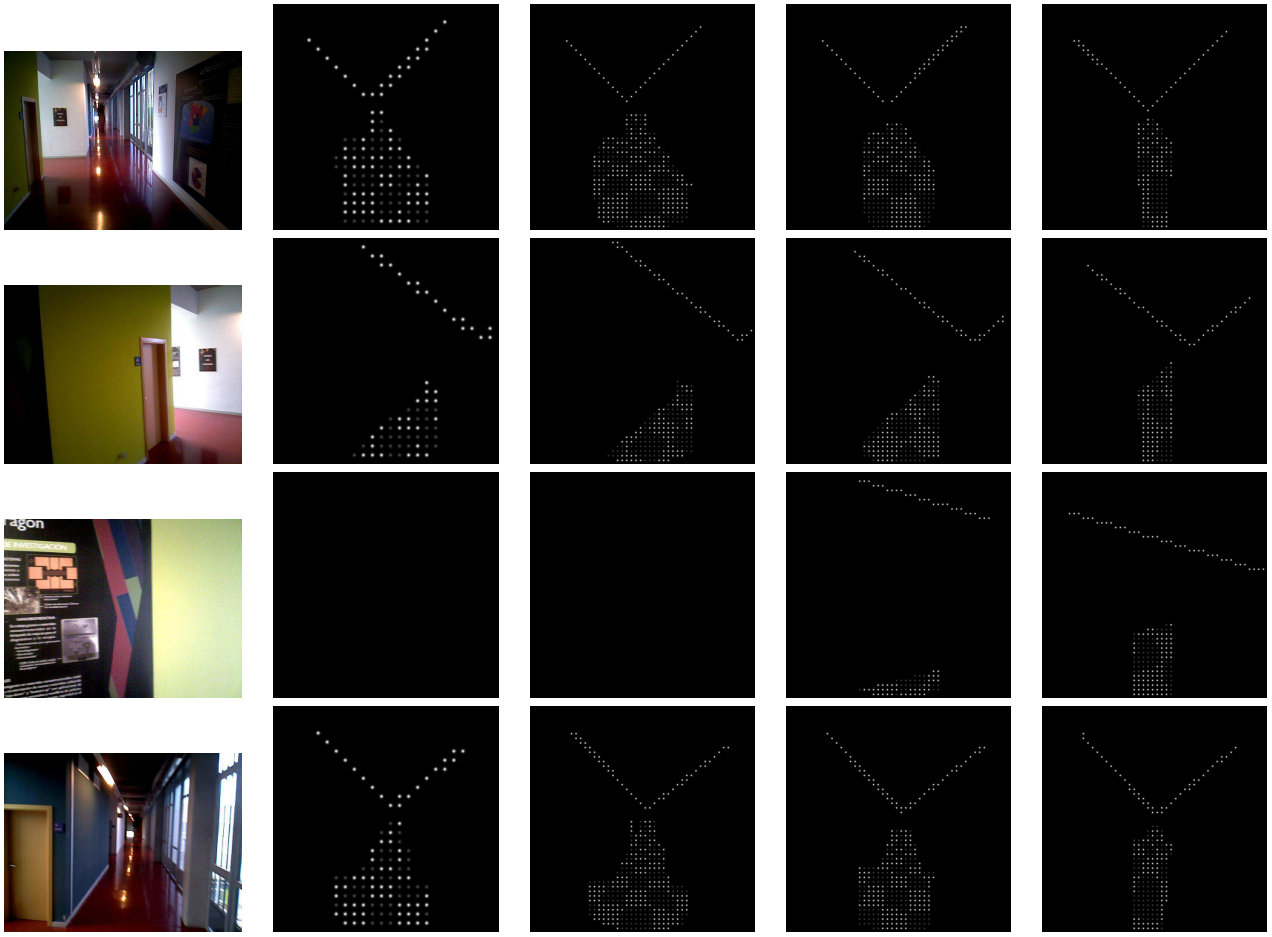

Figure 5: Four real examples of our phosphene-based representation including different parameters. In particular, we wanted to show the difference of using different number of phosphenes (N_p) and field of view of simulated phosphene camera (given by its focal length f). By columns, RGB Image and four alternative phosphenic representations (1–4) with parameters: (1) $f = 525px$, $N_p = 484$; (2) $f = 525px$, $N_p = 1862$; (3) $f = 400px$, $N_p = 1862$; (4) $f = 200px$, $N_p = 1862$.

tive representation would be to keep the pattern white and black. However, then it would be impossible differenciate between obstacle and black squares. In that case we propose to draw the borders of the floor polygon. The ceiling shows two lines of a corridor pointing to the front vanishing point, which was selected as direction to follow.

We have tested different parameters of the codification, including the number of phosphenes and the field of view of the representation. About the first, we can observe in Fig. 5 the differences among columns (1) and (2), where different values of number of phosphenes $N_p = 1862$ and $N_p = 484$ phosphenes. While still useful, it is less intuitive the representation with fewer phosphenes: the chess pattern is not so easily observed, and the bigger discretization in the representation produces sudden *jumps* in the representation in the borders of the obstacles. When the amount of phosphenes increases, the changes in the phosphene map are usually less aggressive and the chess pattern can be clearly observed. We have to note that this is a simulation to show how our approach works for several levels of detail. In a real-world prosthetic system the number of phosphenes would be limited by the current state of technology, which is expected to increase in the future.

The field of view of the representation is another parameter we can tune since we are turning 3D information into a 2D representation. Column (2) shows a representation considering a focal length similar to the RGB-D camera ($f = 525$ pixels). This has the advantage of representing only what is actually viewed by the camera, and provides more accurate delimitation of the obstacles and therefore how to avoid them. However, the field of view of conventional RGB-D cameras is limited, less than normal human

vision. An alternative is shown in column (4), where the field of view selected is very high ($f = 200$ pixels). This has the advantage of showing the information in shorter ranges. For example, when looking at a wall, a low-FoV representation turns all the phosphenes to black (since there are no floor viewed). With a high-FoV representation, a portion of the floor still appears in the image showing the user that he still has some space to move (see third row in Fig. 5). The high-FoV representation has an important drawback, since it needs to encode more information in an already limited display. Note that, in the fourth phosphene map column in Fig. 5, the path seems narrower than with larger values of f. This is because we limit the extension of the free moving polygon to the field of view of the sensor (since it is our only cue to detect obstacles), and it represents less extension relative to the large field of view of the phosphene camera ($f = 200$ pixels). In column (3) we show a middle ground ($f = 400$ pixels), where the information at mid-distance is informative enough and also gives more information about the moving space in short distances.

5.3. Video demonstration

We also have included two videos in this submission, to show the algorithm running in real case scenarios. The videos can also be found online[1]. In particular, we show a video where the user moves in a corridor, and another where the user moves inside an office. In these sequences we use $N_p = 1862$ and $f = 525px$. We can see how the phosphenic representation shows clearly the moving area over which the user can walk safely.

Unlike other works, our representation includes a checkerboard floor which shows the movement of the user in the scene providing a comfortable sense of depth. This is more noticeable in the corridor sequence. In that sequence, the windows at the right part of the corridor sometimes show absence of obstacles since glass remains undetected, returning erroneous floor polygons. Note that the floor gives few valid depth points. However, we can maintain its position with respect to the user with the odometry.

The office sequence, on the other hand, presents a cluttered environment with many obstacles. Our method shows the free moving space in front of the user, removing the tables and other objects from the floor polygon. This scenario is particularly challenging, and thus some frames show undetected portions of obstacles, producing inaccurate polygons. However, these situations occur mostly in isolated frames, producing an effect similar to flickering. Note that, our algorithm for obstacle detection is yet in development and we expect to increase its robustness.

[1] http://webdiis.unizar.es/%7Eglopez/spv.html

6. Conclusions

Visual prostheses are able to evoke visual perception in blind people by using electrical stimuli. However, these prototypes suffer from a lack of spatial and intensity resolution that in practice prevents from transmitting depth perception. In this paper we present an approach to represent depth and motion cues with phosphene patterns in the context of safe navigation of blind people in complex or unfamiliar environments. This approach takes advantage of computer vision algorithms for evoking phosphenes-based stimuli with semantic meaning. In particular, we propose a free-space and obstacles detection algorithm for depicting an iconic representation of a safe navigation layout. The effectiveness of this approach is tested in simulation with real data from indoor environments.

In the near future we expect to perform experiments with people in simulated environments. We expect to collect data from those experiments to support the validity of our method with a quantitative analysis.

Acknowledgments.

This work was supported by Projects DPI2014-61792-EXP and DPI2015-65962-R (MINECO/FEDER, UE) and grant BES-2013-065834 (MINECO).

References

[1] A. Ahuja, J. Dorn, A. Caspi, M. McMahon, G. Dagnelie, L. DaCruz, P. Stanga, M. Humayun, and R. Greenberg. Blind subjects implanted with the Argus II retinal prosthesis are able to improve performance in a spatial-motor task. *British Journal of Ophthalmology*, 95(4):539–543, 4 2011.

[2] A. Aladren, G. Lopez-Nicolas, L. Puig, and J. J. Guerrero. Navigation assistance for the visually impaired using RGB-D sensor with range expansion. *IEEE Systems Journal*, 10(3):922–932, 2016.

[3] L. N. Ayton, C. D. Luu, S. A. Bentley, P. J. Allen, and R. H. Guymer. Image processing for visual prostheses: A clinical perspective. In *IEEE International Conference on Image Processing*, pages 1540–1544, 2013.

[4] N. Barnes. An overview of vision processing in implantable prosthetic vision. In *IEEE International Conference on Image Processing*, pages 1532–1535, Sept 2013.

[5] J. Bermudez-Cameo, A. Badias-Herbera, M. Guerrero-Viu, G. Lopez-Nicolas, and J. J. Guerrero. RGB-D computer vision techniques for simulated prosthetic vision. In *Iberian Conference on Pattern Recognition and Image Analysis*, pages 427–436, 2017.

[6] G. S. Brindley and W. S. Lewin. The sensations produced by electrical stimulation of the visual cortex. *The Journal of Physiology*, 196:479–493, 1968.

[7] K. Cha, D. K. Boman, K. W. Horch, and R. A. Normann. Reading speed with a pixelized vision system. *JOSA A*, 9(5):673–677, 1992.

[8] S. C. Chen, G. J. Suaning, J. W. Morley, and N. H. Lovell. Simulating prosthetic vision: I. visual models of phosphenes. *Vision Research*, 49(12):1493–1506, 2009.

[9] S. C. Chen, G. J. Suaning, J. W. Morley, and N. H. Lovell. Simulating prosthetic vision: II. measuring functional capacity. *Vision research*, 49(19):2329–2343, 2009.

[10] J. M. Coughlan and A. L. Yuille. Manhattan world: Compass direction from a single image by bayesian inference. In *The Proceedings of the Seventh IEEE International Conference on Computer Vision*, volume 2, pages 941–947, 1999.

[11] G. Dagnelie, editor. *Visual Prosthetics*. Ed. Boston, MA: Springer US, 2011.

[12] D. Dakopoulos and N. Bourbakis. Wearable obstacle avoidance electronic travel aids for blind: A survey. *IEEE Trans. on Systems, Man, and Cybernetics, Part C*, 40(1):25–35, 2010.

[13] G. Denis, C. Jouffrais, C. Mailhes, and M. J. Macé. Simulated prosthetic vision: Improving text accessibility with retinal prostheses. In *Annual International Conference of the IEEE Engineering in Medicine and Biology Society*, pages 1719–1722, 2014.

[14] D. Feng and C. McCarthy. Enhancing scene structure in prosthetic vision using iso-disparity contour perturbance maps. In *2013 35th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*, pages 5283–5286, July 2013.

[15] M. A. Fischler and R. C. Bolles. Random Sample Consensus: A paradigm for model fitting with applications to image analysis and automated cartography. *Commun. ACM*, 24(6):381–395, June 1981.

[16] A. Flint, D. Murray, and I. Reid. Manhattan scene understanding using monocular, stereo, and 3d features. In *2011 International Conference on Computer Vision*, pages 2228–2235, Nov 2011.

[17] A. P. Fornos, J. Sommerhalder, and M. Pelizzone. Reading with a simulated 60-channel implant. *Frontiers in neuroscience*, 5, 2011.

[18] C. S. S. Guimaraes, R. V. B. Henriques, and C. E. Pereira. Analysis and design of an embedded system to aid the navigation of the visually impaired. In *2013 ISSNIP Biosignals and Biorobotics Conference: Biosignals and Robotics for Better and Safer Living (BRC)*, pages 1–6, Feb 2013.

[19] D. Gutiérrez-Gómez, W. Mayol-Cuevas, and J. J. Guerrero. Inverse depth for accurate photometric and geometric error minimisation in RGB-D dense visual odometry. In *IEEE International Conference on Robotics and Automation*, pages 83–89, 2015.

[20] V. Hedau, D. Hoiem, and D. Forsyth. Recovering the spatial layout of cluttered rooms. In *2009 IEEE 12th International Conference on Computer Vision*, pages 1849–1856, Sept 2009.

[21] L. Horne, J. Alvarez, C. McCarthy, M. Salzmann, and N. Barnes. Semantic labeling for prosthetic vision. *Computer Vision and Image Understanding*, 149:113–125, 2016.

[22] L. Horne, N. Barnes, C. McCarthy, and X. He. Image segmentation for enhancing symbol recognition in prosthetic vision. In *Annual International Conference of the IEEE Engineering in Medicine and Biology Society*, pages 2792–2795, 2012.

[23] H. Josh, C. Mann, L. Kleeman, and W. L. D. Lui. Psychophysics testing of bionic vision image processing algorithms using an FPGA hatpack. In *2013 IEEE International Conference on Image Processing*, pages 1550–1554, Sept 2013.

[24] J.-H. Jung, D. Aloni, Y. Yitzhaky, and E. Peli. Active confocal imaging for visual prostheses. *Vision research*, 111:182–196, 2015.

[25] F. I. Kiral-Kornek, C. O. Savage, E. O'Sullivan-Greene, A. N. Burkitt, and D. B. Grayden. Embracing the irregular: A patient-specific image processing strategy for visual prostheses. In *Annual International Conference of the IEEE Engineering in Medicine and Biology Society*, pages 3563–3566, 2013.

[26] D. C. Lee, M. Hebert, and T. Kanade. Geometric reasoning for single image structure recovery. In *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pages 2136–2143, June 2009.

[27] W. H. Li. Wearable computer vision systems for a cortical visual prosthesis. In *Workshop on Assistive Computer Vision and Robotics (ACVR) - Proceedings of the IEEE International Conference on Computer Vision Workshops*, pages 428–435, 2013.

[28] W. H. Li. *A Fast and Flexible Computer Vision System for Implanted Visual Prostheses*, pages 686–701. Springer International Publishing, Cham, 2015.

[29] Y. Li, C. McCarthy, and N. Barnes. On just noticeable difference for bionic eye. In *Annual International Conference of the IEEE Engineering in Medicine and Biology Society*, pages 2961–2964, 2012.

[30] W. L. D. Lui, D. Browne, L. Kleeman, T. Drummond, and W. H. Li. Transformative reality: improving bionic vision with robotic sensing. In *Annual International Conference of the IEEE Engineering in Medicine and Biology Society*, pages 304–307, 2012.

[31] C. McCarthy, N. Barnes, and P. Lieby. Ground surface segmentation for navigation with a low resolution visual prosthesis. In *Annual International Conference of the IEEE Engineering in Medicine and Biology Society*, pages 4457–4460, 2011.

[32] C. McCarthy, D. Feng, and N. Barnes. Augmenting intensity to enhance scene structure in prosthetic vision. In *2013 IEEE International Conference on Multimedia and Expo Workshops (ICMEW)*, pages 1–6, July 2013.

[33] C. McCarthy, J. G. Walker, P. Lieby, A. Scott, and N. Barnes. Mobility and low contrast trip hazard avoidance using augmented depth. *Journal of neural engineering*, 12(1):016003, 2014.

[34] H. Meffin. What limits spatial perception with retinal implants? In *IEEE International Conference on Image Processing*, pages 1545–1549, 2013.

[35] R. Oktem, E. Aydin, and N. Cagiltay. An indoor navigation aid designed for visually impaired people. *IECON*, pages 2982–2987, 2008.

[36] B. Peasley and S. Birchfield. Real-time obstacle detection and avoidance in the presence of specular surfaces using

an active 3D sensor. In *IEEE Workshop on Robot Vision (WORV)*, pages 197–202, 2013.

[37] A. Perez-Yus, D. Gutierrez-Gomez, G. Lopez-Nicolas, and J. J. Guerrero. Stairs detection with odometry-aided traversal from a wearable RGB-D camera. *Computer Vision and Image Understanding*, 154:192–205, 2017.

[38] H. Schafer, A. Hach, M. Proetzsch, and K. Berns. 3D obstacle detection and avoidance in vegetated off-road terrain. In *IEEE International Conference on Robotics and Automation*, pages 923–928, 2008.

[39] N. R. Srivastava. *Simulations of Cortical Prosthetic Vision*, pages 355–365. Springer US, Boston, MA, 2011.

[40] A. Stacey, Y. Li, and N. Barnes. A salient information processing system for bionic eye with application to obstacle avoidance. In *Annual International Conference of the IEEE Engineering in Medicine and Biology Society*, pages 5116–5119, 2011.

[41] R. W. Thompson, G. D. Barnett, M. S. Humayun, and G. Dagnelie. Facial recognition using simulated prosthetic pixelized vision. *Investigative ophthalmology & visual science*, 44(11):5035–5042, 2003.

[42] J. J. van Rheede, C. Kennard, and S. L. Hicks. Simulating prosthetic vision: Optimizing the information content of a limited visual display. *Journal of vision*, 10(14):32–32, 2010.

[43] V. Vergnieux, M. J. M. Mac, and C. Jouffrais. Wayfinding with simulated prosthetic vision: Performance comparison with regular and structure-enhanced renderings. In *2014 36th Annual International Conference of the IEEE Engineering in Medicine and Biology Society*, pages 2585–2588, Aug 2014.

[44] H.-C. Wang, R. K. Katzschmann, S. Teng, B. Araki, L. Giarre, and D. Rus. Enabling independent navigation for visually impaired people through a wearable vision-based feedback system. In *IEEE International Conference on Robotics and Automation*, pages 6533–6540, 2017.

[45] J. Wang, X. Wu, Y. Lu, H. Wu, H. Kan, and X. Chai. Face recognition in simulated prosthetic vision: face detection-based image processing strategies. *Journal of neural engineering*, 11(4):046009, 2014.

[46] J. D. Weiland, N. Parikh, V. Pradeep, and G. Medioni. Smart image processing system for retinal prosthesis. In *Annual International Conference of the IEEE Engineering in Medicine and Biology Society*, pages 300–303, 2012.

[47] F. Wong, R. Nagarajan, and S. Yaacob. Application of stereovision in a navigation aid for blind people. *Proceedings of the 2003 Joint Conference of the Fourth International Conference on Information, Communications and Signal Processing, 2003 and Fourth Pacific Rim Conference on Multimedia*, 2:734–737 vol.2, 2003.

[48] M. P. H. Zapf, M.-Y. Boon, N. H. Lovell, and G. J. Suaning. Assistive peripheral phosphene arrays deliver advantages in obstacle avoidance in simulated end-stage retinitis pigmentosa: a virtual-reality study. *Journal of neural engineering*, 13(2):026022, 2016.