IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 21 December 2023, accepted 15 January 2024, date of publication 22 January 2024, date of current version 1 February 2024.

Digital Object Identifier 10.1109/ACCESS.2024.3357400

RESEARCH ARTICLE

RASPV: A Robotics Framework for Augmented Simulated Prosthetic Vision

ALEJANDRO PEREZ-YUS[1], MARIA SANTOS-VILLAFRANCA[1], JULIA TOMAS-BARBA[1], JESUS BERMUDEZ-CAMEO[1], LORENZO MONTANO-OLIVAN[2], GONZALO LOPEZ-NICOLAS[1], (Senior Member, IEEE), AND JOSE J. GUERRERO[1]

[1]Instituto de Investigación en Ingeniería de Aragón (I3A), University of Zaragoza, 50018 Zaragoza, Spain
[2]ITAINNOVA—Instituto Tecnológico de Aragón, 50018 Zaragoza, Spain

Corresponding author: Alejandro Perez-Yus (alperez@unizar.es)

This work was supported by Projects PID2021-125209OB-I00 (funded by MCIN/AEI/10.13039/501100011033 and FEDER/UE), and JIUZ2022-IAR-05. DGA 2022-2026 grant supports Maria Santos-Villafranca.

ABSTRACT One of the main challenges of visual prostheses is to augment the perceived information to improve the experience of its wearers. Given the limited access to implanted patients, in order to facilitate the experimentation of new techniques, this is often evaluated via Simulated Prosthetic Vision (SPV) with sighted people. In this work, we introduce a novel SPV framework and implementation that presents major advantages with respect to previous approaches. First, it is integrated into a robotics framework, which allows us to benefit from a wide range of methods and algorithms from the field (e.g. object recognition, obstacle avoidance, autonomous navigation, deep learning). Second, we go beyond traditional image processing with 3D point clouds processing using an RGB-D camera, allowing us to robustly detect the floor, obstacles and the structure of the scene. Third, it works either with a real camera or in a virtual environment, which gives us endless possibilities for immersive experimentation through a head-mounted display. Fourth, we incorporate a validated temporal phosphene model that replicates time effects into the generation of visual stimuli. Finally, we have proposed, developed and tested several applications within this framework, such as avoiding moving obstacles, providing a general understanding of the scene, staircase detection, helping the subject to navigate an unfamiliar space, and object and person detection. We provide experimental results in real and virtual environments. The code is publicly available at www.github.com/aperezyus/RASPV

INDEX TERMS Computer vision, navigation, RGB-D, simulated prosthetic vision, visually impaired assistance.

I. INTRODUCTION

Although traditional tools to assist the visually impaired such as the cane and the guide dog are robust and cost-effective, they lack the potential to improve the quality of life of the patients to a greater extent by taking advantages of new technologies. Since the 1960s, different research works have found that electrical stimulation of the visual cortex or other parts of the visual pathway (such as the retina) caused patients afflicted by the aforementioned disorders to perceive bright spots of light called *phosphenes*. Nowadays, Visual Prostheses (VPs) are becoming a promising method to treat incurable eye disorders such as Age Macular Degeneration or Retinitis Pigmentosa. VPs generally consist of retinal or cortical implants that apply electrical stimulation using an electrode array to generate a grid of phosphenes. As a matter of fact, experimental results demonstrate that implanted patients were able to develop coordination using their visual prosthetic device improving significantly over residual native vision in spatial-motor tasks [1]. Due to the biological nature of these systems, the generation of these phosphene visualizations is still far from producing fully functional vision, and there are some problems and limitations to overcome [2], [3]. For instance, the resolution of the phosphene grid, (*i.e.* number of phosphenes) is constrained by biology, technology and safety, although this is in continuous evolution and

The associate editor coordinating the review of this manuscript and approving it for publication was Lei Shu.

improving over time [4]. Besides, there are other limitations: the small field of view (FOV), the low brightness range, and typical malfunctions such as dropout, noise, or irregularities in size, shape and color of the phosphenes [5]. In addition, there are also temporal dynamics involved in phosphene elicitation: phosphenes are not instantaneously turned on and off, taking some time to become fully illuminated and fade out [6].

Despite the steady evolution of VPs, the medical procedures and safety requirements cause that usually years of experimentation and clinical trials are needed before new advances reach commercial approval. Although there are other existing models [7], [8], two commercial prosthetic vision systems undoubtedly are representative examples of the two main ways of approaching the whole perception process and communication via VPs: the Argus II from Second Sight [9], and the Alpha-IMS from Retina Implant AG [10], [11]. Both of them, involve more elements than the retinal implant itself. For example, the typical components of the Argus II vision system are: a small camera mounted on the eyeglasses that is used for acquiring images, and a portable computer to convert the image data into an electronically coded signal that is transferred to the implant via wireless communication. On the other hand, Alpha-IMS uses microphotodiode arrays inside the eyeball to capture the light and directly transform it into an electronic signal that is received by the visual prosthesis to elicit phosphenes. The resolution of the Alpha-IMS is much higher in comparison (1500 vs 60 phosphenes), but some research points out that this apparently large numeric distance does not completely correlate to a significant improvement on visual acuity and functionality while performing several tasks [2]. The main reason could be that with the Argus II it is possible to adjust the stimulation variables for each electrode individually, whereas with the Alpha-IMS the control over the stimulation is limited [12]. This finding potentially allows a camera-driven system such as the Argus II to leverage image processing techniques to enhance the stimulation to display patterns or highlight features such as edges or obstacles.

Therefore, besides the clinical research on VPs, nowadays there are important efforts to design and implement algorithms that, despite the limitations of current VPs, manage to effectively communicate the relevant information in the environment and assist in daily life tasks [13]. This is the main focus of our work. The situations where a system able to augment the relevant information could be useful are numerous. For instance, when an implanted patient wants to find and grab a specific object, a camera-based prosthetic vision system could process the camera feed at high resolution and highlight the object so it is easier to grab. Or it could be useful to help a subject to move around in an unfamiliar environment, possibly with moving obstacles. A prosthetic device without any enhancement of this sort may be unable to properly inform the subject correctly, or even put him at risk in certain situations (*e.g.* with obstacles or staircases).

In order to avoid complex and costly trials on real patients, a non-invasive method to evaluate the efficacy of VPs is Simulated Prosthetic Vision (SPV). The idea of SPV is to replicate what subjects with VPs can perceive and represent it on screens or Head-Mounted Displays (HMD), with the objective that individuals with healthy vision can take part in experiments. However, most existing SPV approaches do not consider more advanced perception devices than regular cameras, or the simulation of the prosthetic devices is too rigid and simplified. Besides, temporal dynamics are usually ignored, which diminishes the realism of experiments that require real-time interactions. In general, most existing implementations are too task-specific and specially tailored to perform a particular experiment, which makes it more difficult to recycle the framework for new experimentation.

CONTRIBUTIONS

In this work, we present a novel SPV framework that overcomes several limitations of current systems while being flexible, realistic and with a modular structure that allows adding new features over time. An overview is shown in Fig. 1, which can be used as guiding diagram to follow the explanations of the manuscript and understand the different modules and components. RASPV is developed in a robotics framework, allowing the implementation of algorithms that work in real-time and making it possible to integrate artificial intelligence methods from the robotics and computer vision literature, moving forward towards smart visual prosthesis.

First, the input information of the system is the data acquired by the sensors, which may come from real or virtual cameras in simulated environments. Our system can work both with color and depth cameras, so the acquired information is processed with image or point cloud processing algorithms in order to extract relevant information from the scene. Then, the result of that processing is transformed into a visualization that simulates what real patients can see in our SPV module. It can simulate highly configurable VPs, with modifiable number and size of phosphenes, spatial distribution, levels of luminosity, FOV, and also temporal effects like trails observed in real prostheses, besides other defects such as noise or dropout. The brightness of each phosphene is chosen individually depending on the phosphenic representation mode selected, the available levels of luminosity, and the temporal phosphene model.

We have developed new smart modes of visualization that provide semantic and useful information via iconic representations and augmented reality to assist the subject in different tasks such as obstacle avoidance, object and stair detection, scene recognition, or navigation. This final phosphene image can be visualized with a Head-Mounted Display (HMD) to make the simulation more immersive and allow the user to browse the scene with head movements. The virtual setting allows us to extend the experimentation and perform more controlled and easily quantifiable tests in tasks such as guided navigation [14]. Experiments show

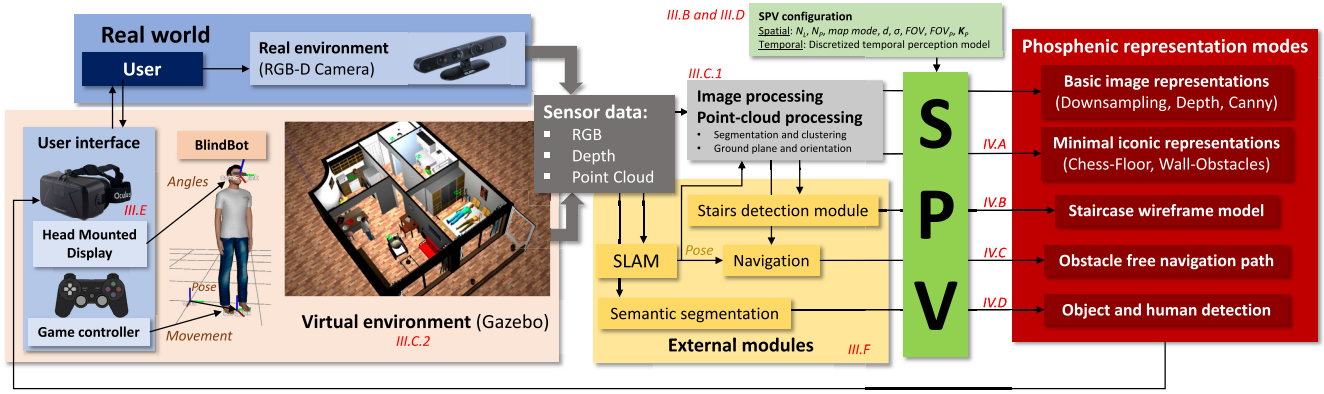


FIGURE 1. Overview of RASPV, including all the main components and modules in different boxes, with the corresponding section in the manuscript written in red. The user may be performing an experiment in a real environment, with a camera, or in simulation, in a virtual environment where the user controls a human avatar with a game controller. In either modality, the sensor data comes from an RGB-D camera, which allows us to safely interact in the 3D environment. The centerpiece of RASPV is the SPV module, which receives the sensor data processed by computer vision algorithms, and possibly the output of some external robotic modules, to produce phosphene images depending on the selected representation mode. The SPV module can be configured to reproduce visual prostheses with realistic spatial and temporal models from the literature. We have developed several new phosphenic representation modes, particularly some that leverage image or point-cloud processing, staircase and object detection, SLAM, or autonomous navigation to provide informative visual cues to the user in order to assist in certain tasks. The resulting phosphene images are visualized by the user with a Head Mounted Display during the experiment.

the performance of the method in both real and virtual configurations. We believe that a modular framework such as RASPV is a relevant and necessary tool for the community since it allows other researchers to focus on designing experiments or novel representations without developing their own SPV implementation, which is a time-consuming and complex task. The code is publicly available at www.github.com/aperezyus/RASPV

II. RELATED WORK

While there is no standard way to implement SPV, Chen et al. discussed extensively how a standardized simulation should be and what parameters a good prosthetic vision simulator should have [15]. Leveraging SPV, we can study what are the minimum requirements of VPs to perform several tasks. This is useful for contextualizing current systems and for analyzing where to focus research on image encoding techniques [16], [17], [18], [19]. Most of the early work is based on basic image processing techniques, such as *downsampling* [13], which is also the method used in the Argus II [9]. Given the low resolution and dynamic range of VPs, this may not be the most useful and robust representation in many situations, especially those that involve hazards such as stairs or risk of collision. There have been other approaches that use advanced computer vision techniques which can enhance the semantics and the relevance of the information to display, developing what we call *phosphenic representation modes* to benefit from different visual functions in different environments [20]. For example, saliency can be used to highlight the presence of obstacles [21], [22], or image segmentation to help distinguish classes [23]. Recently, deep learning has also been used in this field. For example, [24] combines object mask segmentation with structural edges from the scene to create a phosphenic image that is used to evaluate object and room-type identification.

On the other hand, many researchers have used RGB-D instead of conventional RGB cameras because the additional depth (D) channel is particularly useful when addressing mobility and real-world interaction, compared to RGB cameras that may not be as reliable for detecting obstacles if they are not salient enough in the image. For example, McCarthy et al. have proposed RGB-D methods to detect objects that are threats to collisions or to make them more salient [25]. With depth information, a neural network is trained in [26] to detect structural edge information. The work of [27] uses RGB-D to provide a sense of depth and motion while at the same time informing of the presence of obstacles and the orientation of the scene.

Other works take a different approach and use virtual-reality-based environments to evaluate the user response with different models of visual representation [28], [29], [30], [31], [32]. This procedure allows to try new representations and perform extensive tests with people in a realistic manner while reducing the complexity of performing the experiment. Some of these systems, however, use virtual environments just to obtain images that are similar to what conventional cameras can obtain, not allowing to benefit from the usage of new cameras such as RGB-D [30], [31].

While existing SPV systems are suitable to perform certain tasks and experiments, they are often unrealistic or lack the capability to be used in tasks other than the specific trial they were designed for. Some SPV approaches use over-simplified visualization of phosphenes (*i.e.* pixelized [16], [21], rough circles [17], [18], [19], [22]), or ignore defects such as dropout or noise [23], [33], which limits the conclusions drawn by the experimental results.

In a continuous attempt to improve the realism of the simulations of phosphenes, new spatial [34], [35], temporal [6], and spatiotemporal [36], [37] models have been introduced in the last few years based on reports from trials

with implanted patients. Until very recently, most studies simulate phosphenes as small, isolated light spots, arranged regularly in a grid, which has been referred to as *Scoreboard model* [35]. The work from Beyeler et al. [35] shows how the phosphenes are generally perceived as distorted and elongated shapes, and that this seems to be correlated to the activation of the electrodes in the axon fibers, which leads to their proposed *Axon Map model*. This spatial model distorts the shapes of phosphenes following the optic nerve fiber bundles of the subjects recovered from the ophthalmic fundus image. Thus, these distortions are not only patient-specific, but also co-related to the electric pulse parameters. On the other hand, works [6] and [36] introduce mathematical models to simulate the temporal dynamics of the phosphene elicitation. However, the introduction of such complex spatial and temporal models is difficult to apply in a real-time simulator, particularly considering that if several electrodes are stimulated at the same time, the behavior of the electrodes cannot be known simply by linearly combining the independent phosphene perception of each electrode [35].

The implementation from [38] includes some of these models, but it only works by offline rendering a single image or video, taking a long time to process. Additionally, this implementation is adapted for a VR simulator of prosthetic vision in [39]. However, it only introduces the spatial model without temporal dynamics, and the image processing is mostly reduced to edge detection. Finally, the study [40] includes temporal effects in a SPV system, however, it conducts simple reading tasks experiments through virtual reality glasses. The lack of real-time simulators with temporal dynamics that work in both real and virtual environments, facilitating the execution of complex experiments involving everyday tasks, made us design a way to introduce a validated temporal model [6] in our framework.

III. RASPV FRAMEWORK

This section thoroughly describes our framework and all its implemented features (Fig. 1). Particularly, we describe the communication between modules, the simulation of the prosthetic vision, the acquisition of information, the visualization on a Head-Mounted Display (HMD), the introduction of temporal dynamics, and the usage of some smart modules developed from robotics and computer vision state of the art results.

A. COMMUNICATION BETWEEN MODULES

Our system currently has many different modules, and many more could be added over time. The way we handle the communication between different modules is using Robot Operating System (ROS) [41]. ROS is a set of software libraries especially convenient for managing communication in robotic tasks. The information flows via messages, such as the commands to move a robot or the perceptual information retrieved by it. Since ROS is widely used in the robotics community, there is a large number of packages already developed which can be integrated for particular adjacent tasks, such as SLAM (Simultaneous Localization and Mapping) or autonomous navigation. Using ROS offers flexibility since one module could be easily substituted by another without significant changes of the system. It is programmed in C++ in real-time.

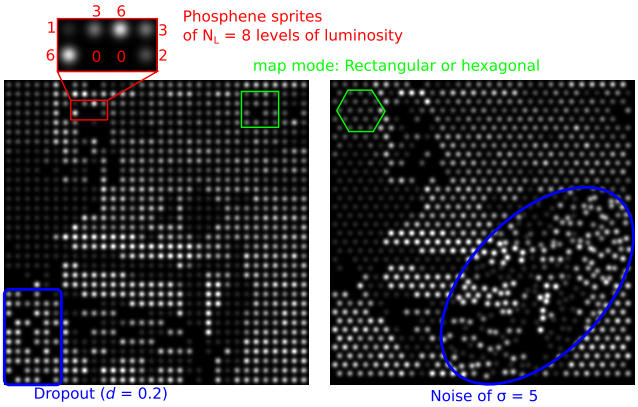


FIGURE 2. Two phosphene images obtained from RASPV with approximately 1000 phosphenes, showing the main features and parameters of our SPV module.

B. SPATIAL SIMULATION OF PHOSPHENES

For our SPV framework, we pursue to create a realistic simulation of what a patient with VPs can perceive while keeping the simulation parametrized and flexible to be easily adapted to existing VPs or future models. A detailed description of this simulation is included in the Appendix A. To summarize, the main points considered on the system configuration of our SPV module are the following:

1) PHOSPHENE APPEARANCE

A phosphene is represented as a circular white dot with a Gaussian profile so that maximum intensity is reached in the center and progressively dims as the radius increases. This kind of representation is widespread in the literature and is general enough to remove patient-specific spatial deformations. Nevertheless, given the system's modularity, other models could also be implemented [35]. It is possible to elicit phosphenes at different luminosity levels [15] (Fig. 2).

2) PHOSPHENE MAPPING

Since phosphenes are elicited using a grid of electrodes, it produces a perception of individual phosphenes spatially arranged in a regular pattern in the FOV of the patient. Our configuration parameters are the number of phosphenes and the *map mode*, or lattice of the grid depending on the electrodes (rectangular or hexagonal, see Fig. 2). Implementation-wise, the black image is automatically filled with phosphenes arranged according to the *map mode* selected and the aspect ratio, which leads us to several phosphene center positions in the image P_i. The resulting image is the phosphene image $\mathcal{I}$ (Fig. 3).

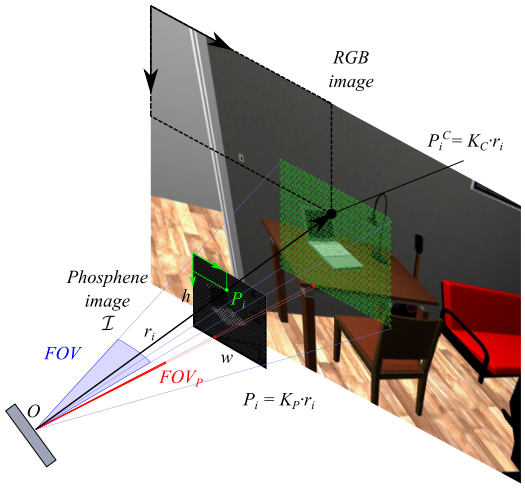


FIGURE 3. Projective geometry relating camera and phosphene images, including the field of view of the prosthesis (*FOV*) and the size of the phosphene (FOV_P) parameters.

3) DEFECTS ON THE MAP

Reports point out that the spatial distribution of phosphenes is subject to patient-specific deviations. Thus, for the visualization of phosphenes, we include a *drop-out rate* parameter (d), that randomly shuts down phosphenes (*i.e.* zero luminosity), and a σ parameter that models the standard deviation of the position P_i with respect to the computed mapping (Fig. 2).

4) FIELD OF VIEW AND PHOSPHENE SIZE

The field of view (FOV) parameter is frequently given for each visual prosthesis. Particularly, we define a calibration matrix $\mathbf{K}_P$ (assuming pin-hole camera model) that, for each point in the phosphene image (P_i), can be used to trace a ray r_i (Fig. 3) that allows us to enable person-environment interaction with the 3D world perceived by the cameras. Similarly, the size of the phosphenes is usually given in arcs of the field of view (parameter FOV_P).

C. ACQUISITION OF INFORMATION

Sensor data may be acquired with real cameras in the real world or with virtual cameras in a simulated environment. In order to interact in the environment safely and efficiently, we propose to use RGB-D cameras as the main sensor which, besides color images, also provide point clouds that enable reasoning in the 3D space (Section III-C1). Next, in Section III-C2, we detail our virtual environment implementation.

1) IMAGE AND POINT CLOUD PROCESSING

In our problem, a direct encoding between the camera image and the phosphene image is not adequate since we need to consider the different FOVs (Fig. 3). For each phosphene position P_i computed for the phosphene image, we compute the equivalent phosphene position in the camera image with $P_i^C = \mathbf{K}_C \cdot \mathbf{K}_P^{-1} \cdot P_i$, where P_i and P_i^C are in homogeneous coordinates. Note that since the FOV of prosthetic systems is typically smaller than the FOV of the camera, all P_i^C will be inside the camera image. With this approach, we can implement basic visualization modes that directly encode image information to phosphenes. The simplest one would be *downsampling*: grayscale values in the pixel positions P_i^C are proportionally converted to the N_L luminosity levels of our system. Some pre-processing, such as edge detection or semantic segmentation, could be applied to enhance the information before encoding. For instance, in [24], both object detection and layout estimation are used to enhance the visualization in scene understanding tasks with phosphenes. Depth measurements from a depth image could be directly mapped to the N_L luminosity levels as well, *e.g.* making the values closer to the camera appear brighter to alert of collisions.

One of our main goals is to overcome the limitations of existing VPs by performing advanced 3D processing with point clouds [42] and augment the relevant information of the scene with iconic representations that do not need to rely so much on high resolution phosphene images. The basic 3D features that we represent on the phosphene images can be reduced to points, lines and planes. For points (and clusters of points), we directly project 3D coordinates of the 3D points to the phosphene grid with $\mathbf{K}_P$ and find the closest phosphene in the image. Similarly, lines are projected onto the image plane and phosphenes within a pixel distance to the line are selected. For planes, we intersect the rays from each phosphene r_i with the 3D plane to obtain the intersection point.

Estimating the ground plane (*i.e.* gravity direction and distance to ground) is essential for many applications discussed in this work since it helps to orient and contextualize the subject in the scene. After a planar segmentation [42], we choose the most likely ground plane out of the segmented planes in the scene, considering size, distance to the camera, and orientation. The transformation ${}^C T_F$ is then computed by aligning the y axis of the new reference frame F with the ground plane normal n_f (Fig. 4). Additionally, the remaining two *Manhattan World* directions [43] can be recovered considering the best alignment with the normals of the rest of the scene, following [44].

2) VIRTUAL ENVIRONMENT

Using simulated environments for SPV experimentation is important to reduce costs and risks, as well as to simplify some real-world implementation difficulties, allowing researchers to focus on the parts that actually need evaluation. It also allows us to systematically perform experiments in equal conditions to a wider range of subjects, with accurate measurements of all actions performed during the test. In our system, we can recover the pose of the subject at all times and thus have a clear understanding of the trajectory, the time that takes the subject to perform some task, as well

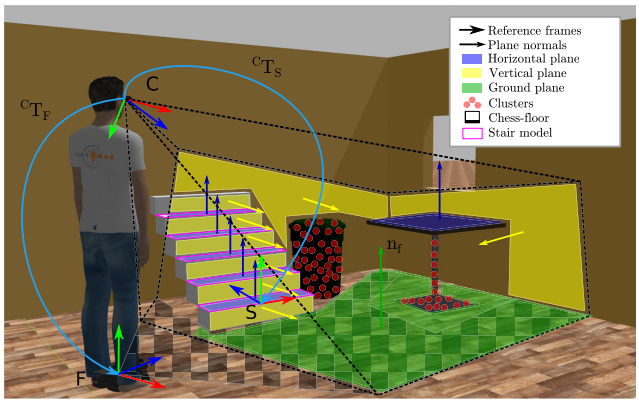


FIGURE 4. Visual example of the main elements that take part in the 3D processing of point clouds carried out in the Minimal iconic representations and Stair detection.

as gathering other task-specific data such as counting the number of collisions during a mobility-related experiment.

For our framework, we use *Gazebo* [45], a robotics simulator that includes realistic robot and sensor models that were designed to test algorithm implementations. Combining ROS with Gazebo is perfectly suitable for our needs: we can use a simulated RGB-D camera in Gazebo (*e.g.* an Asus Xtion Pro Live), and pass the information captured in the virtual environment to our perception module using the message system from ROS. The same SPV implementation would be compatible straight away when a real camera is used instead of a Gazebo simulation since they use the same type of ROS messages.

In Gazebo, it is possible to create worlds by adding elements that also have physical properties (*i.e.* mass, inertia, collisions). It is possible as well to import complex models of buildings and objects made with other software (such as *Sweet Home 3D* [46], see Fig. 1), which allow us to propose enough variety for our experiments, something very costly in the real world.

In order to move and interact inside the virtual environment, we have implemented a new robot model called *BlindBot*, based on Turtlebot, that simulates a human subject. The robot's base includes two wheels and differential drive movement, as well as several sensors such as bumpers (to detect collisions) and an RGB-D camera (Asus Xtion Pro Live or an Intel RealSense R200). We included a human model and placed the RGB-D camera at the eyes position, replicating the idea of the glasses-mounted camera of real systems (see Fig. 1). Regarding the user interface, controlling the robot's movement can be done with either a game controller or a keyboard, allowing to turn the base and to move forwards and backwards (or combine both), as well as to change linear or angular velocities. To make the robot's movements more realistic, we added some additional joints between the camera link and the base link to make it possible to change the view frame similarly to what humans do when moving the head. This allows to decouple the view from the movement of the whole robot which allows to explore the scenes more naturally. In particular, we add three cylindrical joints corresponding to roll-pitch-yaw angles. It is also relevant to mention that it is possible to recover the pose of the robot with respect to the virtual world reference frame (Fig. 1) so that we do not need to use any localization method or any ground plane extraction algorithm.

D. SIMULATION OF TEMPORAL DYNAMICS

Previous studies have shown that retinal ganglion cell stimulation produces visual percepts consisting of an initial brightening and subsequent fading until phosphenes are turned off [47]. Therefore, in order to make the SPV system as realistic as possible, it is important to integrate these temporal effects described in the literature by prosthesis users. This will not only allow us to study their impact on performing several tasks, but also to propose palliative strategies to such effects considering a human-in-the-loop scheme, as in [40].

Usually, temporal dynamics are disregarded in related works due to the difficulties of introducing them into a real-time implementation. However, there are validated existing models based on experiments with real patients that can be used to simulate the phosphene illumination over time. We choose to use the temporal model of [6], for its consistency with the biological nature of neural systems, and because this model has been used in other works such as [37] to describe the appearance of phosphenes. Our approach could be extended to other similar models as well [36].

Experiments from [6] reveal that the response time of the visual stimulus is about 200 ms, much larger than the temporal changes that vision is able to perceive (roughly 20 ms). Using their mathematical model, we can simulate the illumination dynamics of each individual phosphene depending on the input electric signal (*e.g.* pulses, train of pulses). In order to choose realistic simulation parameters, we use the average patient-specific parameters from [6], and the same type of electrical stimulus as with Retina Implant AG's Alpha-AMS implant (*i.e.* biphasic pulses). For each input pulse, the model outputs a signal representing the illumination level of the phosphene over time: it illuminates fairly fast, but the decay of the pulse is slower and generates a visual trail sensation. However, if several pulses are produced in a row, the behavior is not linear and requires expensive calculations at a very small sampling time with respect to the working frequencies, particularly considering these dynamics need to be implemented in a simulator that works in discrete time.

In order to implement it in real-time, we propose a *discretized temporal perception model* that stores the pre-computed luminosity value of the phosphene according to the information obtained in previous instants (Fig. 5). First, the illumination results have been computed using the continuous perception model from [6] for all possible combinations that may have occurred in the current frame and the previous frames. The input stimuli are computed

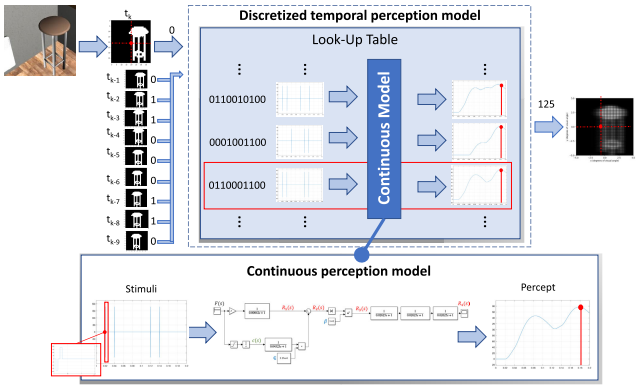


FIGURE 5. General scheme of the Discretized temporal perception model implemented in RASPV. For each phosphene, we take the input signal sequence from the current and previous 9 frames, which consists of biphasic pulses or no pulses regarding the output of the visualization mode (in the figure, binarization of a stool). The continuous model [6] predicts the percept signal and, particularly, the current luminosity to show in SPV at this time instant. In order to make this problem feasible in real-time, we have pre-computed all possible combinations of input stimuli in a Look-Up Table.

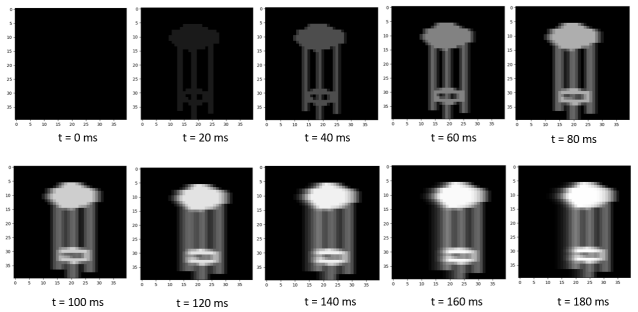


FIGURE 6. Output of the temporary model of the stool moving to the right.

for each phosphene depending on the output of the visual representation module so that a biphasic pulse happens if this module outputs 1 for the pixel corresponding to that phosphene position (*e.g.* a binarized image that highlights some object). Alternatively, no pulse happens at that time instant if the module outputs 0 in that pixel. The final illumination values for each pulse combination are stored in a Look-Up Table whose input index is a sequence of ones and zeros depending on the current and previous frames. Once this information is computed for all possible combinations, the maximum value obtained will have the maximum brightness value and the rest will be scaled according to this maximum. This pre-computation of phosphene illuminations with respect to current and past stimuli allows us to run the temporal model in real-time, covering all plausible stimuli combinations in a 200 ms time-frame. The result of applying the above discrete model to a binarized image of a stool moving to the right is shown in Fig. 6.

E. VISUALIZATION WITH A HEAD-MOUNTED DISPLAY

A head-mounted display (HMD) can be used along our SPV module to test the phosphene visualization in an immersive

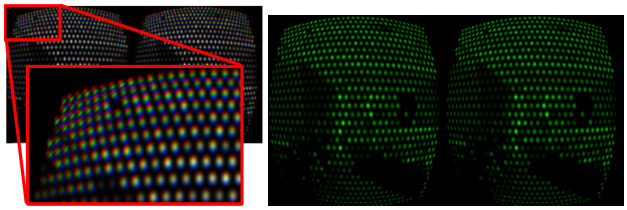


FIGURE 7. In the HMD, images corresponding to the left and right eye are projected on the display. The particular aspheric-distortion strategy used for an adequate focus of the image produces color artifacts when using white phosphenes (a), avoided when using a single color channel (b).

experience. In particular, we use an Oculus Rift DK2, which estimates the head's orientation based on an Inertial Measurement Unit (IMU). There are differences between the representation used on the screen and the one used in the HMD. The HMD uses a set of aspheric lenses to correctly focus the displayed image at such a short distance to the user's eyes. As a result of that, the image is distorted to compensate the barrel effect. We realized that the Bayer color pattern is not perfectly mapped by the lenses and, as a consequence, it is not possible to represent white phosphenes in the periphery of the image without color artifacts (Fig. 7a). Since color is unimportant in a phosphene representation, we use a single color channel in the head-mounted representation. In particular, we use the green channel, which has the highest resolution in the Bayer pattern (Fig. 7b).

In order to connect the HMD to our system, our current implementation uses two separate computers connected via a socket. The *server* performs all computations of RASPV, whereas the *client* has the HMD connected and performs two main tasks: receive the phosphenic representation to be shown in the display, and send the 3D orientation of the HMD to Gazebo in order to move the camera in the BlindBot model. The communications have been handled with a ROS node that sends the angles from the IMU to the Gazebo model and the camera moves right away without significant lag. Therefore, the subject can move the head naturally during the experiments while using the controller to navigate. This implementation is fast enough for real-time (communication takes less than 100 ms), and since most powerful computations are performed in the server, a more lightweight processing unit could be used to connect the HMD to improve portability.

F. EXTERNAL ROBOTIC MODULES

One of the advantages of using ROS to centralize the communication among modules is that we can re-purpose packages from the robotics field into our framework. Here, the most relevant ones are described:

1) STAIR DETECTION

Stairs can be a dangerous structure for the visually impaired, being a potential source of accidents. Here, we use the method proposed in [44] and [48], which uses RGB-D cameras and

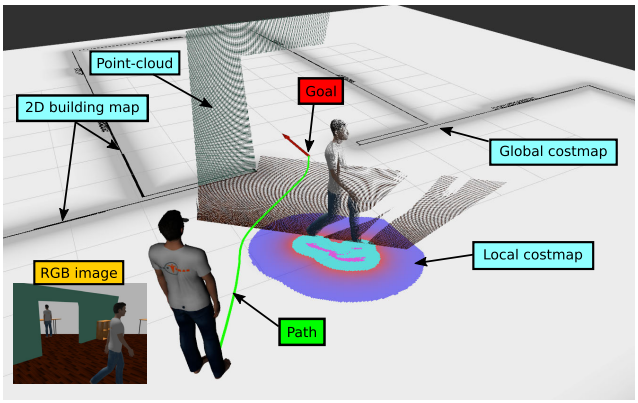


FIGURE 8. The navigation module computes the shortest path towards a predetermined goal (in the figure, a door passage). The path is computed considering a known 2D map of the building (which introduces a global costmap) and the point cloud from the RGB-D camera (which we use to detect obstacles and dynamically update the local costmap and thus the shortest path to reach the goal). In this figure, we can observe that the path (in green) is curved to avoid collision with a walking person.

thus is straightforward to integrate into our system. This method recovers the pose of the stair with respect to the camera (${}^{C}T_{S}$) and the measurements of every step, therefore providing the complete stair model (Fig. 4), useful not only to alert the subject but also for guidance.

2) SLAM

Many applications require to keep track of the followed path to be able to come back to a particular place, for example to reach a relevant instance previously detected. Sometimes, it is necessary to know the subject's position on the map in order to give proper indications to reach a destination. To handle these problems, we have used a SLAM (*Simultaneous Localization and Mapping*) method that is compatible with ROS and RGB-D cameras [49]. It allows us to know our relative position with respect to the initial frame since it keeps the system localized in a map while it is being built online. Besides, if the environment has been previously mapped, it is possible to launch in *localization mode*, which returns the absolute position of the subject in the map.

3) AUTONOMOUS NAVIGATION

The objective of a navigation system is to find a path from a starting point to a goal point. This is one of the most obvious tasks for visually impaired assistance, and has been extensively studied in the past [50]. Such a task involves perception, path-planning, and collision avoidance. We used the standard ROS *navigation* package [41]. In our framework, the global path-planner is based on the A* algorithms, which provide the shortest distance trajectory from a starting point to a goal destination given a known map or a building floor plan. There is also a local planning based on Dynamic Window Approach (DWA) [51] that provides obstacle avoidance by computing the optimal collision-free velocity. Two costmaps are used by the path-planner: the global path-planner uses a costmap based on the map of the environment, and the local planner updates a dynamic costmap using the depth information. Both costmaps are grids in which a cost is assigned depending on the obstacle size and distance. Fig. 8 shows an example of the local and global costmaps with the corresponding computed path. In our system, we only need the camera to solve localization (using SLAM) and obstacle avoidance, not needing any other external devices like a cane [52] or wearables [53].

4) SEMANTIC SEGMENTATION

One of the most useful ways to assist a visually impaired with computer vision is via object detection/recognition. We propose to use deep learning-based semantic segmentation approaches since they allow us to recognize objects of interest and locate them in the image in order to highlight them and help the user, for example, to reach and grasp the object. We developed a ROS node that launches a Pytorch implementation of a pre-trained semantic segmentation network DilatedNet [54], [55]. Particularly, the architecture using Resnet50Dilated as the encoder and PPM deepsup as the decoder. It was trained in the ADE20K dataset [56], [57], [58], which contains 150 object classes, achieving a good balance in accuracy and running time when adjusted to 3 iterations (about 0.25 *s* per frame including node communications).

IV. AUGMENTED REPRESENTATION MODES

The different modules of RASPV described in Section III serve as foundation to design new *phosphenic visualization modes* that allow the patients to retrieve more information from the visual prosthesis, considering their strong limitations. Then, to test the effectiveness of the methods, the next step would be to perform experiments with sighted people, which would allow us to extract conclusions about the validity of said modes. In this work, our main motivation lies mostly on mobility-related issues, *i.e.* to provide helpful information to move safely, efficiently and purposely, for instance, by providing depth and motion cues, orientation, the path to follow, or even some rough understanding on the scene. Having depth information is particularly useful in this context since it allows us to have structural information of the environment, including walls, floor, obstacles, and other particular instances, such as stairs.

Nevertheless, we cannot lose focus on the fact that the perception with visual prosthesis has severe limitations. Therefore, our proposed representations attempt to be minimal (just using two or three levels of phosphene brightness) and based on iconic and easily recognizable cues. Besides, to convey additional information such as the presence of staircases, objects, or a path to follow, we propose to use augmented reality-based representations that highlight and superimpose the additional useful information over the current visualization mode. In the next sections, we show our main proposals as well as some results in virtual and real environments. The video attached to this submission as

supplementary material shows some of these results in real sequences.

A. MINIMAL ICONIC REPRESENTATIONS

Basic representations such as downsampling or direct depth-mapping may not be effective or safe enough given the limitations of the VPs. Here, we propose minimal iconic representations that leverage information recovered from point clouds and try to represent the scene so that basic structures and elements can be easily recognized. The aim of these iconic representations is to transmit confidence to the user so that it can move around the environment avoiding collisions while providing additional informative cues. The two main minimal iconic representations implemented in RASPV are Chess-Floor (first introduced in [27]) and Wall-Obstacles:

1) CHESS-FLOOR

The idea of this minimal representation is to bring a sense of depth, orientation and movement using as few luminosity levels as possible. Inspired by old low resolution video-games, we propose to add texture to the ground, particularly with black and white squares (like a chessboard), which can be implemented in phosphene representation with gray and white phosphenes. The SLAM module allows the user to move freely while the texture remains fixed on the floor to effectively transmit the depth and motion cues. Additionally, the texture can be set to match the orientation of the scene by recovering the three main directions [43]. To enhance the iconic representation, we also propose to display the textured floor only in obstacle-free areas, so that it simultaneously serves as an obstacle avoidance system. This can be achieved with an RGB-D camera, since from the point clouds it is possible to detect shapes and surfaces that could block the user's movement. By turning the ground phosphenes off where the obstacles are detected with the RGB-D camera, the representation allows for safe and comfortable navigation. In Fig. 4 we show a visual example of a scene with the chess polygon drawn inside the obstacle-free area in front of the subject.

2) WALL-OBSTACLES

The goal of this representation is to assist in navigation tasks by roughly displaying the main elements on the scene (*e.g.* planes, objects). With the point clouds from the RGB-D camera, we can segment the scene in planes (by applying region-growing methods) and clusters of points (by Euclidean cluster extraction) [42]. The planes orthogonal to the ground are classified tentatively as *walls* (*e.g.* yellow planes in Fig. 4). In the phosphene image, the *wall* planes are shown in gray, and may be useful to detect the boundaries of the room or big furniture. Alternatively, horizontal planes (*e.g.* blue plane in Fig. 4) and clusters of points (which may account for non-planar instances, *e.g.* red points in Fig. 4) are shown in white, to highlight them over the background. The floor remains black with turned off phosphenes, so that all the objects and planes, which may be additional hazards, are more easily identifiable.

FIGURE 9. Sample frames of different phosphenic representations obtained from a sequence in a virtual environment. Our proposed minimal iconic representations (Chess-Floor and Wall-Obstacles) are compared to basic image processing alternatives (*i.e.* grayscale downsampling, and depth scaling), showing that we can convey relevant information for mobility with just a few levels of luminosity.

In order to show the effectiveness of our minimal iconic representations, we compare them with other basic methods that directly process RGB and depth images. In Fig. 9 we show four frames of a sequence, where the BlindBot walks through a corridor and finally enters the living room. At the top, we show the RGB images with the positions of the phosphenes in green circles. Notice that we are simulating a prosthesis with large FOV (around 47°) in order to improve the visibility of the figures. In successive rows, we show all the representations.

The Chess-Floor representation is able to transmit a sense of depth and movement inside the scene thanks to the checkerboard pattern. Since it only draws walkable areas, it is easy not to crash with any obstacle and it shows when the floor extends in new rooms, as we can see in the first frame of Fig. 9. Nevertheless, it is not very informative once we are in the room we intend to be, since it does not show any of the objects. In contrast, Wall-Obstacles representation is able to convey a rough understanding of the scene and allows to distinguish some objects, although relies on the subject's own memory and ability to navigate to prevent collisions.

On the other hand, the basic representations have some limitations regarding mobility. For example, downsampling has the huge limitation of being unable to distinguish between

different instances with similar gray levels, which may result in trouble detecting hazards and obstacles. Depth information seems to work reasonably well in that regard since it is directly translating the proximity of every pixel to the subject. Nevertheless, we take into account that these images are taken in simulation, and, therefore, the depth perception has perfect accuracy, without noise or missing pixels as happens in real life. Considering real cases of VPs with limitations such as noise and dropout, the basic representations may turn extremely confusing, whereas a more rough approach such as our iconic representations could be more robust to such occurrences.

We have also evaluated the Chess-Floor representation in real settings. We have recorded several sequences of indoor environments, and run our algorithm. For these experiments, we have used the visual odometry method from [59]. We have recorded two sequences, *corridor* and *office*, whose videos can be found in the supplementary video.

B. STAIRS DETECTION

Stairways are structures that are common to all human-made scenes, but also a potential risk of accidents. Therefore, it is important for users to detect them in the environment. To display stairs in RASPV, when an instance of a staircase is detected, the segments of the 3D wireframe model of the staircase (*e.g.* pink lines in Fig. 4) are projected to the phosphene view. To highlight the presence of the staircase in the view of the subject, we apply maximum luminosity to the lines of the stair, and dim the luminosity of the rest of phosphenes. Therefore, the stairs detection module works as augmented reality, and it is complementary to any representation mode. The goal of this implementation is to make a method to alert the user of hazards, but also to be informative about the presence of staircases regardless of the visualization mode currently selected.

We have included staircases in a Gazebo map and run our implementation with different representations. In Fig. 10 we show two examples, with an ascending and a descending staircase, respectively. Our method clearly highlights the presence of staircases, even pointing out the surface of the step where the subject can step on. Since the rest of the image is dimmed in intensity, the stair is easy to detect. In the case of the downsampling representation, it happens often that the color difference between the floor and the steps is very small, and thus the risk of having an accident is high. Having the stair detector and stair augmentation active, reduces such risk since it alerts with enough time.

For the stairs detection in real settings, we tested several sequences provided by [44] (Fig. 11). In the first example, we can see how the system is able to inform the subject of the presence of both ascending and descending staircases from a large distance, also recovering the full pose and thus enabling the possibility of guiding the user to face the staircase straight or close to the handrails. The second example shows the user approaching the ascending staircase and up to four steps

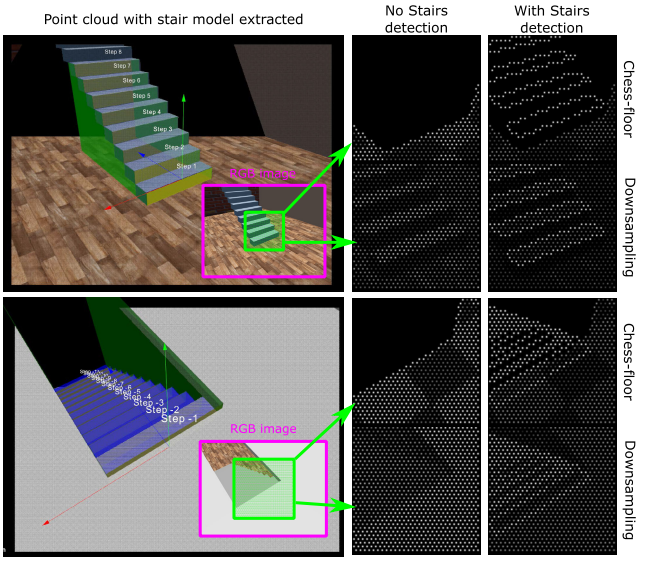


FIGURE 10. The stair detection and augmentation method from RASPV are shown in two frames with, respectively, an ascending and descending stair. For each frame, the point cloud with the resulting staircase from [44] is shown on the left, with the corresponding RGB image and the field of view of the simulated prosthetic device marked in green. On the right, the phosphenic visualization of those frames with and without the stair augmentation. Results are shown with two representations as well: Chess-Floor [27] and downsampling.

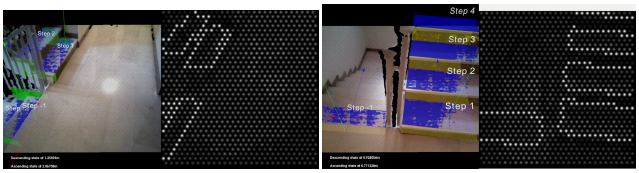


FIGURE 11. Two frames from a sequence with stairs, where we show our stairs detection application included in RASPV. On the left, point cloud with 3D staircase model overlayed. On the right, the corresponding SPV image.

correspondingly shown in the phosphene visualization. As we can see with real images, this system could help prevent subjects from having an accident but also be very helpful to efficiently traverse a building.

C. NAVIGATION ASSISTANCE

As mentioned above, it is crucial for users to navigate autonomously in the environment. Therefore, we propose this augmented mode, which highlights the path, to assist users with VP in reaching their destination in unfamiliar settings, such as public spaces (*e.g.* hospitals, stations, shopping malls). To represent the path returned by the navigation module (presented in Section III-F) in the phosphene image, we project the path as a set of 3D line segments. To avoid confusion or information overload, only the current visible path is shown, since we can use depth information from the camera to hide occluded segments of the path.

For the navigation experiments, we have created several virtual environments that include conventional elements such as tables, doors or human beings. The 2D map used for

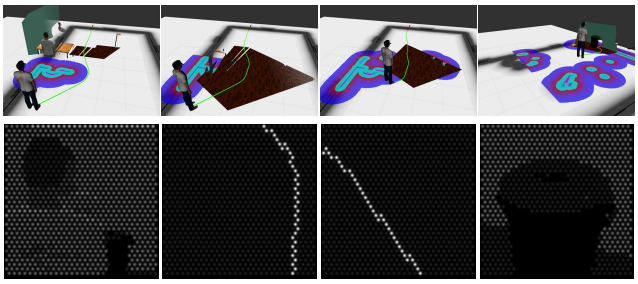


FIGURE 12. This experiment shows a sequence where the subject is asked to find a bin in order to dispose of some garbage. On the first row, the navigation assistant with all the elements as shown in Fig. 8. On the bottom row, the corresponding view from the SPV module presented in this work.

navigation only marks the occupancy of the structural and permanent elements in the scene (*e.g.* floor plan of the building), and the rest of the elements (obstacles) will be dynamically recovered by the navigation assistant, and can be easily obtained with the RGB-D camera given the pose is known in Gazebo. In Fig. 12 we show four frames of a sequence where the subject is asked to navigate towards a goal. As we can observe, initially the subject is only able to perceive a person in front of him. Since the trajectory is overlayed on the ground, like an augmented reality navigation assistant, the subject just needs to find the path and follow the trail. The obstacles are detected and informed to the navigation assistant, which steers the subject away from them, while at the same time keeping the shortest possible path. It is important to notice that the FOV of the camera is typically larger than the FOV of the VP, so the path is safe enough to follow and avoid any collision.

For the real world experiment, we have recorded a long sequence inside a building previously mapped with a 2D laser scan. However, for the navigation, it is necessary to have the current location of the subject at all times. Since the 2D building map provided was captured from another type of sensor, it is not possible to directly localize using the information from the head-mounted RGB-D camera. Therefore, we proceeded in two steps: First, mapping with the RGB-D SLAM method from [49], and then running the method in localization mode to get the absolute pose. Between both operations, we manually aligned the 3D feature map to the 2D building map provided, so that the position of the subject on the map can be recovered and included in the navigation assistant. In Fig. 13 we show several frames from our experiment. Notice that the representation mode chosen does not draw phosphenes in the floor, which makes a perfect pairing with this navigation assistant. The longer sequence is also included as supplementary material in the form of a video. We also show an additional experiment with dynamic objects such as a person walking by. There we can see how the path changes as the obstacle is detected and then recalculates again when the obstacle disappears, always keeping the shortest possible path. Notice that the subject is also moving towards the goal, increasing the complexity of the sequence even more.

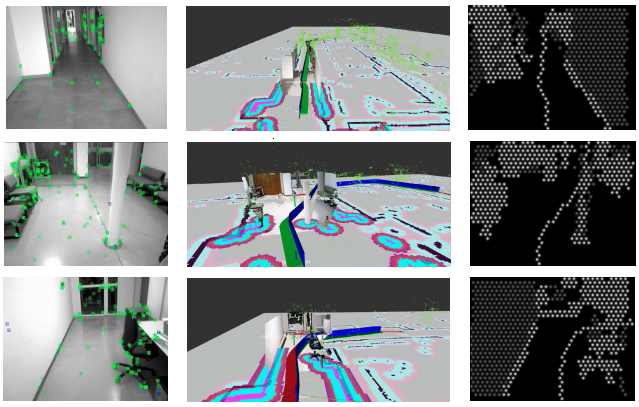


FIGURE 13. Several frames from the sequence in a real environment, where we show the SPV image with an augmentation mode that overlays the guiding path over a representation that only shows vertical planes in gray and horizontal planes and clusters of points in white (right). For the navigation, we use a SLAM system [49] (left, feature points in green). At the center, we show the navigation information.

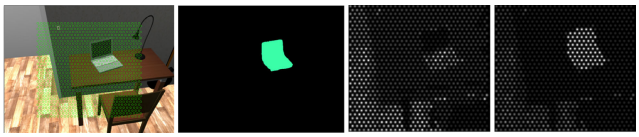


FIGURE 14. From left to right: RGB image with the green grid showing the phosphene positions. Prediction of the semantic segmentation network. Phosphene map with downsampling mode. Phosphene map with our object detection active, highlighting the location of the desired object.

D. OBJECT AND HUMAN DETECTION

Sometimes, so much information in the scene may be confusing, making the recognition of objects a challenging task. For this reason, we have also implemented visualization modes to convey semantic information to the users. In particular, we also take an augmented reality approach, where we highlight the objects of interest in the scene with the predicted segmentation mask (see Semantic segmentation module in Section III-F) by simply setting the corresponding phosphenes to the maximum level of luminosity, dimming the rest of the image. In Fig. 14 we show an example, where the object of interest is a laptop, and the background representation mode is downsampling. Notice that, when the object detection mode is active, the object can be precisely located and it would help the subject to reach it and grab it, which is not so obvious with the background representation. Only one object is chosen to be highlighted in this case in order to not overwhelm the subject. The object classes to highlight can be adjusted to the necessities of the task.

In Fig. 15 we show some results where, in this case, the desired instance to highlight is a human, also using the semantic segmentation network from before. Here, the temporal dynamics are also active, and it can be observed how the phosphenes take some time to illuminate and

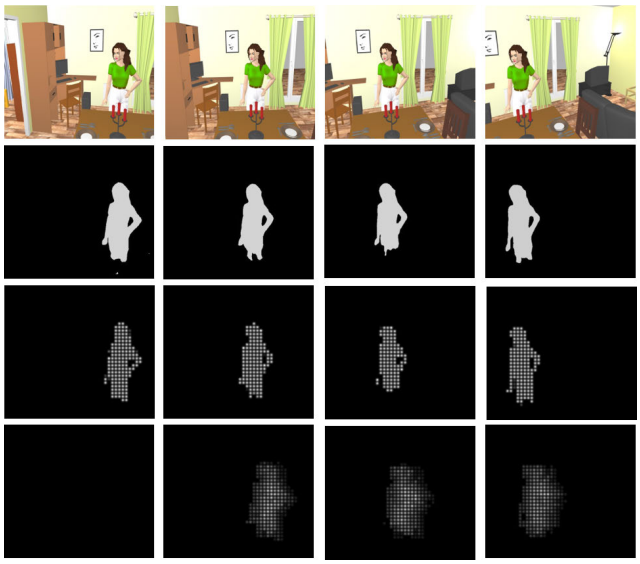


FIGURE 15. Several frames when turning with object detection mode active (here, highlighting a person). Each row represents, respectively: RGB image, segmentation mask, object detection mode without temporal effects, and object detection with temporal dynamics. The temporal dynamics are being simulated with our discretized temporal perception model, which translates to visual trails.

fade out, leaving some trailing effect in the visualization. In the supplementary video, some more examples have been included with the HMD running in the simulator.

V. CONCLUSION AND FUTURE WORK

In this work, we have proposed a novel framework for Simulated Prosthetic Vision developed in an open-source robotics framework that enables real-time performance and integration with external robotics and computer vision algorithms. It provides an easily customizable spatial and temporal model based on the state of the art that simulates what is perceived with a visual prosthesis. The input data comes either from a real RGB-D camera and also from a virtual environment. Using a real camera set-up allows us to validate the feasibility of the different algorithms, considering the technical complexity and real-time restrictions of a real prototype. On the other hand, the option of virtual immersive environments facilitates systematic and repeatable experiments of SPV for the evaluation of different phosphenic representation modes with statistical meaning. Here, we also present several visualization mode proposals, using obstacle detection, ground-plane detection, stairs and object detection, SLAM, and path planning with global and reactive navigation, providing a set of new augmented representations of scenes, staircases, objects and obstacle-free guidance path.

Despite the complexity of the developed system, there are limitations to be addressed in future work, that would improve the usefulness of our SPV system. The realism of the SPV system could be improved following works such as [38], by introducing new spatial [35] and temporal models [40], [60]. Furthermore, user interaction with the SPV system is limited to head movements and the gamepad. Adding touch-based interactions could enhance the patient experience. In addition, integrating multimodal feedback to the user (*e.g.* in the form of audio) has proven to be effective in other applications for visually impaired people.

The main objective of this study is to achieve a realistic SPV system that allows to draw conclusions extensible to real visual prostheses, and to share the code with the whole community so that researchers can use it to carry out their own designed experiments. Although RASPV has already been used for evaluating the navigation module [14], obtaining conclusions from experiments with healthy sighted people, there is still many other possible ways to utilize the proposed system to obtain more clinically meaningful conclusions with statistical significance. In the future, more experiments involving the augmented representation modes presented in this article could evaluate their actual usefulness. For instance, temporal dynamics could be studied by performing similar experiments to those presented in [40]. These experiments measure the influence of temporal effects, including, in our case, a virtual or real environment allowing the subject to look around using virtual reality glasses. Furthermore, following the study of [32], researchers can assess the influence of various parameters of the visual prosthesis using a more realistic simulator, in order to enhance the design of existing ones.

APPENDIX A PHOSPHENE MAPS IN DETAIL

In this appendix, we intend to describe thoroughly the implementation of the simulation of phosphenes of our SPV module, particularly the spatial model we have included in our framework. To represent the output phosphene visualization, first we consider that the entire visual field of the prosthetic device is represented with a 2D image $\mathcal{I}$ of $w \times h$ pixels, where initially all pixel values are set to 0 (*i.e.* entirely black). The phosphenes, whose appearance is described in Section A-A, will be inserted at specific locations of the visual field corresponding to the pixel locations from the phosphene map that is created depending on the specifications of the system (Section A-B). Some typical malfunctions we have considered in our implementation are commented in Section A-C. Finally, we talk about how the phosphene image $\mathcal{I}$ maps the visual field is dealt with in Section A-D.

A. PHOSPHENE APPEARANCE

The most widespread description of the visual perception of a phosphene in the literature is similar to a small, round and colored spot of light in the visual field [15]. However, there are studies that show that it is not as simple as that, since there might be variations in shape (*e.g. doughnut-shaped* [61], elongated [62]), or even color (*e.g.* yellow or blue [61]). Some more recent studies correlate spatial deformations with the nerve fiber bundles of the subject, and the parameters of the electric signal [35]. Nevertheless, these possible variations in shape are difficult to predict and highly patient-specific, and

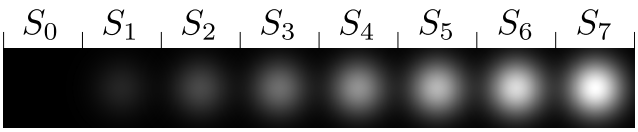


FIGURE 16. Visualization of the phosphene sprites that would compose the phosphene image. In this case, the number of luminosity levels is $N_L = 8$. The last sprite, $S_{N_L} = S_7$, is the sprite of reference S_r. The other sprites at i-th position are created by multiplying each pixel value by $i/(N_L - 1)$.

the perceived shape of the phosphenes cannot be purposely elicited [63] and thus use it to improve representations. Therefore, in our framework, we decided to adopt the most widespread and patient-agnostic phosphene representation, consisting in a round spot of light. Regarding color, white (colorless) seems to be the most common description [61], with no evidence of control over that parameter [15], which is also our color of choice.

We call *phosphene sprite* to the visual representation of a single phosphene, which will be the smallest and more basic element in the whole representation. We adopt the term *sprite* from computer graphics, meaning "a 2D bitmap that is inserted in a larger image", which mimics well how we operate with phosphenes in our framework. For representing the phosphenes, we create a reference phosphene sprite S_r as a small image representing the desired visual representation, *i.e.* a round white spot of light. To make the phosphene sprites more realistic-looking, a two-dimensional Gaussian profile is often used [64] so that the center represents maximum intensity and smoothly dims down its value as the radius increases. Besides, the usage of Gaussian profiles allows to seamlessly address overlap and fuse two adjacent phosphenes by simply summing the pixel values.

While features such as color and shape are not controllable by the prosthetic device, it has been reported that increasing the current through the particular electrode elicits a phosphene of higher intensity or larger size, often correlated [15], [61], [63]. To model this characteristic in our system, we use the parameter N_L to determine the number of discrete levels L of luminance and size. We generate a set $\mathcal{S}$ of N_L phosphene sprites, $\mathcal{S} = \{S_0, .., S_{N_L-1}\}$, which are computed based on a *reference sprite* S_r. The reference sprite is a prototypical phosphene of maximum intensity (the center of the Gaussian of maximum value, *e.g.* 255 for 8-bit images) and size (see Section A-D). The elements $S_i \in \mathcal{S}$ are computed as $S_i = \frac{i}{N_l-1} \cdot S_r$, so that the sprite S_0 will be all zero (thus black, like a shutdown phosphene) and the last sprite in the set $S_{N_L-1} = S_r$. In-between sprites adopt intermediate (gray) values, simultaneously reducing size and maximum intensity from S_{N_L-1} to S_0. In Fig. 16 some sample sprites with $N_L = 8$ are shown.

The number of levels N_L is a parameter of the system. Previous works suggest as a good approximation using 8-16 levels of intensity [15], and we use 8 as default. Nevertheless, it is not clear how feasible it is to actually be able to accurately recognize modulations of intensity. Since it could be hard to discern among adjacent levels, our proposed representations are designed to use as few levels as possible, using instead iconic representations to provide informative perception without relying too much on having high N_L.

B. PHOSPHENE MAP

We use the term *phosphene map* to refer to the spatial distribution of the group of individual phosphenes as displayed in the field of view of the patient. Clinical and biological proofs indicate that phosphene patterns are irregular and patient-specific [65], [66]. Reports have shown that the position of phosphenes in the visual field mostly correlates with the expected region stimulated, especially in retinal implants [61]. While the distribution of phosphenes is always irregular (with almost unrecognizable lattices in cortical implants), in retinal implants the locations are generally consistent with the position of the electrodes in the retina, and thus considering regular distributions is reasonable from a practical standpoint. To provide more realistic output, stochastic jitter offsets should be applied to the exact pixel positions of the phosphenes [15] (more on that in Section A-C).

In the literature, for optical prosthetic devices, the most common phosphene map distributions are rectangular and hexagonal (notice that, in other kinds of prosthetic systems, *e.g.* cochlear, the distributions may be very different). According to [67], a hexagonal map allows a more compact structure and thus, higher density, a desirable property in prosthetic vision. The phosphene mapping consists in computing the pixel positions $P_i = (x_i, y_i)$ in the phosphene image $\mathcal{I}$ where the sprites $\mathcal{S}$ will be located depending on the specifications of the prosthetic system. We have to take into account the following parameters: the number of phosphenes (N_P), the mode of distribution (rectangular or hexagonal) and the size of the image (w, h). Notice that the *aspect ratio* of the visualization is given by the size of the image $w \times h$. For example, Argus II has a phosphene distribution of 10×6. Thus, we would choose a value of $h = 6/10 \cdot w$ for the image $\mathcal{I}$.

Initially, we assume a rectangular grid occupying the whole image and then add variants to the map. To construct the phosphene map, first, we compute the separation δ (in pixels) between the center of consecutive phosphenes (vertically and horizontally). To compute that δ value in the initial case:

$$\delta = \left\lfloor \sqrt{\frac{w \cdot h}{N_P}} \right\rfloor \tag{1}$$

where $\lfloor \cdot \rfloor$ means *floor* operation. We can observe an example of this case at Fig. 17a.

The separation between phosphenes vertically and horizontally can be set differently as δ_x and δ_y, and consider in this particular case that we chose $\delta_x = \delta_y$. For hexagonal grids, by its definition, $\delta_y = \sqrt{3}/2 \cdot \delta_x$. The deltas are then

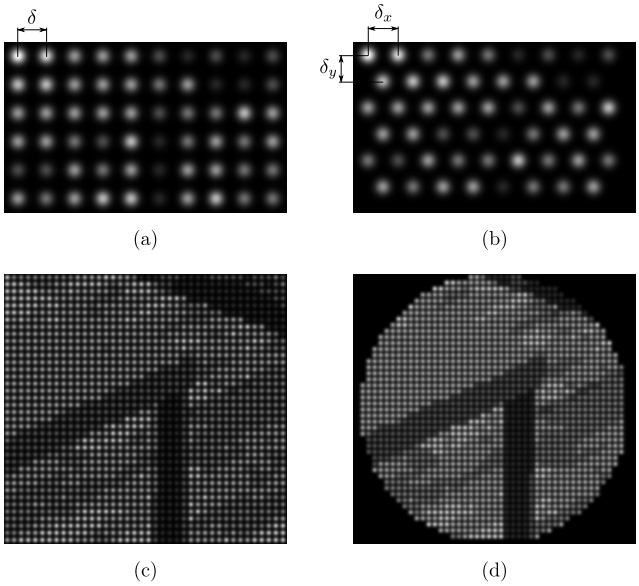


FIGURE 17. Examples of phosphene maps inspired by Argus II (above, with $N_P = 60$ phosphenes) and Alpha-IMS (below, with $N_P = 1500$ phosphenes): (a) Rectangular map. (b) Hexagonal map. (c) Rectangular visual field. (d) Circular visual field.

computed as:

$$\delta_x = \left\lfloor \sqrt{\frac{w \cdot h \cdot 2/\sqrt{3}}{N_P}} \right\rfloor, \quad \delta_y = \left\lfloor \sqrt{3}/2 \cdot \delta_x \right\rfloor \tag{2}$$

In this case, every other row is displaced to the right $\delta_x/2$ pixels (*i.e.* second row starts with $y = \delta_x$). See Fig. 17b for an example.

We also contemplate the case where the visual does not correspond to the whole image but instead a centered circular region. An example with rectangular and circular visual field are shown in Fig. 17c and Fig. 17d. The computation of the deltas needs to consider that, in this case, the area to display phosphenes is not $w \cdot h$. For simplicity, we will assume $\mathcal{I}$ is a square ($w = h$), and the circle is inscribed in the square (radius $= w/2$). Thus, the area of the circle is $\pi/4$ times the area of the square ($w \cdot h$). To get the deltas (for hexagonal grids):

$$\delta_x = \left\lfloor \sqrt{\frac{w \cdot h \cdot 2/\sqrt{3} \cdot \pi/4}{N_P}} \right\rfloor, \quad \delta_y = \left\lfloor \sqrt{3}/2 \cdot \delta_x \right\rfloor \tag{3}$$

In all those situations, we start setting the first phosphene at $P_1 = (\delta_x/2, \delta_y/2)$ and start adding phosphenes side by side following δ_x until the row ends. The next row starts δ_y pixels down, with an extra displacement of $\delta_x/2$ in the case of hexagonal grids. When the circular region mode is selected, we additionally need to check if these phosphenes are inside the circle. Notice that, in all these phosphene maps, the final number of phosphenes might be slightly different than the parameter N_P since we prioritize displaying complete regular grids that cover all the image and some values of N_P may be impossible to reproduce. Nevertheless, this procedure provides a good approximation to the parameter N_P number, and the numeric difference is normally less noticeable than an incomplete grid.

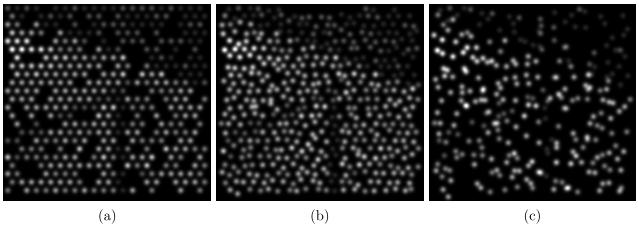


FIGURE 18. Simulations of the alterations of regular maps that occur in real devices. (a) 20% of drop-out ($d = 0.2$). (b) Noise of standard deviation $\sigma = 5$. (c) Both alterations: $d = 0.4$, $\sigma = 10$.

C. ALTERATIONS ON THE MAP: DROP-OUT, NOISE

Numerous reports point out several issues that may arise and are unavoidable at this point due to biological and technological constraints. Two of the most common phosphene map deviations have been included in our SPV module: the drop-out and noise in phosphene positions.

- **Drop-out:** Usually, not all elicited phosphenes are actually turned on [17], [18]. It can happen that even though the electrode emits the signal, the prosthesis malfunctions due to damage in the optical nerve or other parts of the visual system. We address this issue with a parameter of the system called *drop-out ratio*, d, with values between 0 and 1, that will automatically leave $d \cdot N_P$ phosphenes completely shutdown. The phosphenes are chosen randomly and only once at the beginning of the simulation, considering the location of the malfunction is fixed for each individual. An example is shown in Fig. 18a.
- **Noise:** As mentioned in Section A-B, the map is always irregular. To model this, we add some Gaussian noise to the phosphene positions P_i. Thus, we generate new phosphene positions for visualization, $\hat{P}_i$. To get the deviations of $\hat{P}_i$ with respect to P_i, for the normal distribution, we use mean zero and noise standard deviation σ, which is another system parameter. An example is shown in Fig. 18b.

Notice that, since we are trying to replicate the real devices, the subject perceives the dropout and noise effects but they are unknown to the image processing, so it will only be applied for visualization but will not change the positions of the phosphenes P_i in the simulator when they are used to generate phosphene representations. An example of a visualization with high drop-out and noise can be observed in Fig. 18c.

D. FIELD OF VIEW AND PHOSPHENE SIZE

Another important parameter of the prosthetic system is the field of view. While humans on standard conditions have around 210° × 150° of horizontal × vertical field of view that is not achievable in current prosthetic vision devices,

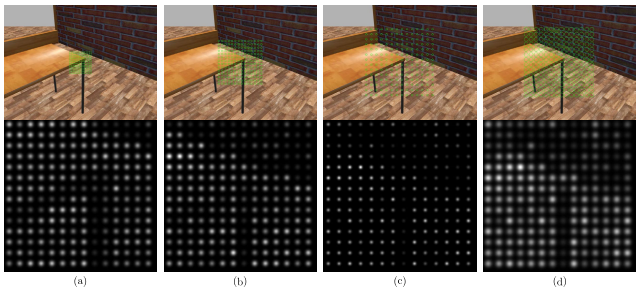


FIGURE 19. Several configurations of our SPV module varying fields of view and size of the phosphenes. For each case, above there is an overlay of the phosphene visual field on the RGB image (of 57° × 43°), and below the corresponding phosphene image. (a) $FOV = 10°$, $FOV_P = 0.5°$. (b) $FOV = 20°$, $FOV_P = 1°$. (c) $FOV = 30°$, $FOV_P = 1°$. (d) $FOV = 30°$, $FOV_P = 2°$.

where fields of view range from $\approx 10°$ to $20°$. Besides, the field of view of cameras is normally different (much larger) than that of the visual prosthesis, so the direct encoding of images to phosphenic representation is inaccurate and may yield to compressed visualizations in the reduced field of view of the subject. This could be problematic or disorienting when interacting with the real world since the position of the perceived features the user might want to interact with will be out of place with respect to their location in the real world and with respect the possible residual vision the patient might have.

To incorporate the system's field of view as a parameter (that in fact, depends on the prosthetic device and is decoupled from the camera), we use the pinhole camera model in the phosphene image. Thus, the same way any camera C has its own intrinsic calibration matrix $\mathbf{K}_C$, we define another intrinsic calibration matrix for the phosphene image $\mathcal{I}$ that we call $\mathbf{K}_P$, and is defined by:

$$\mathbf{K}_P = \begin{bmatrix} f_x^P & 0 & c_x \\ 0 & f_y^P & c_y \\ 0 & 0 & 1 \end{bmatrix} \tag{4}$$

where f_x^P and f_y^P are the focal lengths of the phosphene camera, and $(c_x, c_y) = (w/2, h/2)$ is the center of the image $\mathcal{I}$. To get the focal lengths of the system given the image size $(w \times h)$ and the field of view (FOV) of the prosthetic system, we can use the following formula (showing example for horizontal):

$$f_x^P = \frac{w}{2 \cdot tan(FOV/2)} \tag{5}$$

Similarly, f_y^P can be computed, although we will assume perfectly square pixels and thus $f_x^P = f_y^P = f^P$ for simplicity. Cases where the FOV is different in x and in y are simply modeled by the *aspect ratio* parameter.

This sort of reasoning should apply to any instance that needs to be drawn in the SPV module, particularly phosphenes. In order to replicate prosthetic vision systems, when we talk about *phosphene size*, it should not be described in pixels since real devices do not use pixels. Instead, the size of the phosphenes (particularly the diameter) is given as arcs of field of view (denoted as FOV_P). For example, in [15], it is noted that most reported data regarding the size of phosphenes say they are between 0.5 to 2 degrees of FOV_P. Hence, given the focal length of the system, we can obtain the size of the diameter of the phosphene ϕ_P and thus the size in pixels of the phosphene sprite (width and height) as follows:

$$\phi_P = 2 \cdot f^P \cdot tan(FOV_P/2) \tag{6}$$

The phosphene sprites S_i will be reshaped to be $\phi_P \times \phi_P$ before its insertion in the corresponding phosphene positions in the phosphene map in $\mathcal{I}$. In Fig. 19 there are four examples of different configurations of our SPV module with different FOV and FOV_P.

REFERENCES

[1] A. K. Ahuja, J. D. Dorn, A. Caspi, M. J. McMahon, G. Dagnelie, L. daCruz, P. Stanga, M. S. Humayun, and R. J. Greenberg, "Blind subjects implanted with the Argus II retinal prosthesis are able to improve performance in a spatial-motor task," *Brit. J. Ophthalmol.*, vol. 95, no. 4, pp. 539–543, Apr. 2011.

[2] H. Meffin, "What limits spatial perception with retinal implants?" in *Proc. IEEE Int. Conf. Image Process.*, Sep. 2013, pp. 1545–1549.

[3] M. Beyeler, A. Rokem, G. M. Boynton, and I. Fine, "Learning to see again: Biological constraints on cortical plasticity and the implications for sight restoration technologies," *J. Neural Eng.*, vol. 14, no. 5, Oct. 2017, Art. no. 051003.

[4] S. Ha, M. L. Khraiche, A. Akinin, Y. Jing, S. Damle, Y. Kuang, S. Bauchner, Y.-H. Lo, W. R. Freeman, G. A. Silva, and G. Cauwenberghs, "Towards high-resolution retinal prostheses with direct optical addressing and inductive telemetry," *J. Neural Eng.*, vol. 13, no. 5, Oct. 2016, Art. no. 056008.

[5] G. Dagnelie, "Visual prosthetics 2006: Assessment and expectations," *Expert Rev. Med. Devices*, vol. 3, no. 3, pp. 315–325, 2006.

[6] A. Horsager, S. H. Greenwald, J. D. Weiland, M. S. Humayun, R. J. Greenberg, M. J. Mcmahon, G. M. Boynton, and I. Fine, "Predicting visual sensitivity in retinal prosthesis patients," *Investigative Opthalmol. Vis. Sci.*, vol. 50, no. 4, p. 1483, Apr. 2009.

[7] D. Palanker, Y. Le Mer, S. Mohand-Said, M. Muqit, and J. A. Sahel, "Photovoltaic restoration of central vision in atrophic age-related macular degeneration," *Ophthalmology*, vol. 127, no. 8, pp. 1097–1104, Aug. 2020.

[8] L. Karapanos, C. J. Abbott, L. N. Ayton, M. Kolic, M. B. McGuinness, E. K. Baglin, S. A. Titchener, J. Kvansakul, D. Johnson, W. G. Kentler, N. Barnes, D. A. X. Nayagam, P. J. Allen, and M. A. Petoe, "Functional vision in the real-world environment with a second-generation (44-channel) suprachoroidal retinal prosthesis," *Transl. Vis. Sci. Technol.*, vol. 10, no. 10, p. 7, Aug. 2021.

[9] M. S. Humayun, J. D. Dorn, L. da Cruz, G. Dagnelie, J.-A. Sahel, P. E. Stanga, A. V. Cideciyan, J. L. Duncan, D. Eliott, E. Filley, A. C. Ho, A. Santos, A. B. Safran, A. Arditi, L. V. Del Priore, and R. J. Greenberg, "Interim results from the international trial of second Sight's visual prosthesis," *Ophthalmology*, vol. 119, no. 4, pp. 779–788, Apr. 2012.

[10] A. Rothermel, L. Liu, N. P. Aryan, M. Fischer, J. Wuenschmann, S. Kibbel, and A. Harscher, "A CMOS chip with active pixel array and specific test features for subretinal implantation," *IEEE J. Solid-State Circuits*, vol. 44, no. 1, pp. 290–300, Jan. 2009.

[11] E. Zrenner, K. U. Bartz-Schmidt, H. Benav, D. Besch, A. Bruckmann, V.-P. Gabel, F. Gekeler, U. Greppmaier, A. Harscher, S. Kibbel, J. Koch, A. Kusnyerik, T. Peters, K. Stingl, H. Sachs, A. Stett, P. Szurman, B. Wilhelm, and R. Wilke, "Subretinal electronic chips allow blind patients to read letters and combine them to words," *Proc. Roy. Soc. B, Biol. Sci.*, vol. 278, no. 1711, pp. 1489–1497, May 2011.

[12] H. C. Stronks and G. Dagnelie, "The functional performance of the Argus II retinal prosthesis," *Expert Rev. Med. Devices*, vol. 11, no. 1, pp. 23–30, Jan. 2014.

[13] N. Barnes, "An overview of vision processing in implantable prosthetic vision," in *Proc. IEEE Int. Conf. Image Process.*, Sep. 2013, pp. 1532–1535.

[14] M. Sanchez-Garcia, A. Perez-Yus, R. Martinez-Cantin, and J. J. Guerrero, "Augmented reality navigation system for visual prosthesis," 2021, *arXiv:2109.14957*.

[15] S. C. Chen, G. J. Suaning, J. W. Morley, and N. H. Lovell, "Simulating prosthetic vision: I. visual models of phosphenes," *Vis. Res.*, vol. 49, no. 12, pp. 1493–1506, Jun. 2009.

[16] J. Sommerhalder, B. Rappaz, R. de Haller, A. P. Fornos, A. B. Safran, and M. Pelizzone, "Simulation of artificial vision: II. eccentric reading of full-page text and the learning of this task," *Vis. Res.*, vol. 44, no. 14, pp. 1693–1706, Jun. 2004.

[17] G. Dagnelie, D. Barnett, M. S. Humayun, and R. W. Thompson, "Paragraph text reading using a pixelized prosthetic vision simulator: Parameter dependence and task learning in free-viewing conditions," *Investigative Opthalmol. Vis. Sci.*, vol. 47, no. 3, p. 1241, Mar. 2006.

[18] R. W. Thompson, G. D. Barnett, M. S. Humayun, and G. Dagnelie, "Facial recognition using simulated prosthetic pixelized vision," *Investigative Opthalmol. Vis. Sci.*, vol. 44, no. 11, p. 5035, Nov. 2003.

[19] Y. Zhao, Y. Lu, Y. Tian, L. Li, Q. Ren, and X. Chai, "Image processing based recognition of images with a limited number of pixels using simulated prosthetic vision," *Inf. Sci.*, vol. 180, no. 16, pp. 2915–2924, Aug. 2010.

[20] 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 *Proc. IEEE Int. Conf. Image Process.*, Sep. 2013, pp. 1540–1544.

[21] A. Stacey, Y. Li, and N. Barnes, "A salient information processing system for bionic eye with application to obstacle avoidance," in *Proc. Annu. Int. Conf. IEEE Eng. Med. Biol. Soc.*, Aug. 2011, pp. 5116–5119.

[22] N. Parikh, L. Itti, M. Humayun, and J. Weiland, "Performance of visually guided tasks using simulated prosthetic vision and saliency-based cues," *J. Neural Eng.*, vol. 10, no. 2, Apr. 2013, Art. no. 026017.

[23] L. Horne, J. Alvarez, C. McCarthy, M. Salzmann, and N. Barnes, "Semantic labeling for prosthetic vision," *Comput. Vis. Image Understand.*, vol. 149, pp. 113–125, Aug. 2016.

[24] M. Sanchez-Garcia, R. Martinez-Cantin, and J. J. Guerrero, "Semantic and structural image segmentation for prosthetic vision," *PLoS ONE*, vol. 15, no. 1, Jan. 2020, Art. no. e0227677.

[25] C. McCarthy, J. G. Walker, P. Lieby, A. Scott, and N. Barnes, "Mobility and low contrast trip hazard avoidance using augmented depth," *J. Neural Eng.*, vol. 12, no. 1, Feb. 2015, Art. no. 016003.

[26] D. Feng, N. Barnes, and S. You, "DSD: Depth structural descriptor for edge-based assistive navigation," in *Proc. IEEE Int. Conf. Comput. Vis. Workshops (ICCVW)*, Oct. 2017, pp. 1536–1544.

[27] A. Perez-Yus, J. Bermudez-Cameo, J. J. Guerrero, and G. Lopez-Nicolas, "Depth and motion cues with phosphene patterns for prosthetic vision," in *Proc. IEEE Int. Conf. Comput. Vis. Workshops (ICCVW)*, Oct. 2017, pp. 1516–1525.

[28] G. Dagnelie, P. Keane, V. Narla, L. Yang, J. Weiland, and M. Humayun, "Real and virtual mobility performance in simulated prosthetic vision," *J. Neural Eng.*, vol. 4, no. 1, pp. S92–S101, Mar. 2007.

[29] V. Vergnieux, M. J.-M. Macé, and C. Jouffrais, "Wayfinding with simulated prosthetic vision: Performance comparison with regular and structure-enhanced renderings," in *Proc. 36th Annu. Int. Conf. IEEE Eng. Med. Biol. Soc.*, Aug. 2014, pp. 2585–2588.

[30] 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," *J. Neural Eng.*, vol. 13, no. 2, Apr. 2016, Art. no. 026022.

[31] Y. Zhao, X. Geng, Q. Li, G. Jiang, Y. Gu, and X. Lv, "Recognition of a virtual scene via simulated prosthetic vision," *Frontiers Bioeng. Biotechnol.*, vol. 5, p. 58, Oct. 2017.

[32] M. Sanchez-Garcia, R. Martinez-Cantin, J. Bermudez-Cameo, and J. J. Guerrero, "Influence of field of view in visual prostheses design: Analysis with a VR system," *J. Neural Eng.*, vol. 17, no. 5, Oct. 2020, Art. no. 056002.

[33] T. Han, H. Li, Q. Lyu, Y. Zeng, and X. Chai, "Object recognition based on a foreground extraction method under simulated prosthetic vision," in *Proc. Int. Symp. Bioelectronics Bioinf. (ISBB)*, Oct. 2015, pp. 172–175.

[34] D. Nanduri, M. S. Humayun, R. J. Greenberg, M. J. Mcmahon, and J. D. Weiland, "Retinal prosthesis phosphene shape analysis," in *Proc. 30th Annu. Int. Conf. IEEE Eng. Med. Biol. Soc.*, Aug. 2008, pp. 1785–1788.

[35] M. Beyeler, D. Nanduri, J. D. Weiland, A. Rokem, G. M. Boynton, and I. Fine, "A model of ganglion axon pathways accounts for percepts elicited by retinal implants," *Sci. Rep.*, vol. 9, no. 1, pp. 1–16, Jun. 2019.

[36] D. Nanduri, I. Fine, A. Horsager, G. M. Boynton, M. S. Humayun, R. J. Greenberg, and J. D. Weiland, "Frequency and amplitude modulation have different effects on the percepts elicited by retinal stimulation," *Investigative Opthalmol. Vis. Sci.*, vol. 53, no. 1, p. 205, Jan. 2012.

[37] J. Granley and M. Beyeler, "A computational model of phosphene appearance for epiretinal prostheses," in *Proc. 43rd Annu. Int. Conf. IEEE Eng. Med. Biol. Soc. (EMBC)*, Nov. 2021, pp. 4477–4481.

[38] M. Beyeler, G. M. Boynton, I. Fine, and A. Rokem, "pulse2percept: A Python-based simulation framework for bionic vision," in *Proc. 16th Python Sci. Conf.*, 2017, pp. 81–88.

[39] J. Kasowski and M. Beyeler, "Immersive virtual reality simulations of bionic vision," in *Proc. Augmented Hum. Int. Conf.*, 2022, pp. 82–93.

[40] J. T. Thorn, N. A. L. Chenais, S. Hinrichs, M. Chatelain, and D. Ghezzi, "Virtual reality validation of naturalistic modulation strategies to counteract fading in retinal stimulation," *J. Neural Eng.*, vol. 19, no. 2, Apr. 2022, Art. no. 026016.

[41] Open Robotics. *ROS: Robot Operating System*. Accessed: Jan. 24, 2024. [Online]. Available: https://www.ros.org/

[42] R. B. Rusu and S. Cousins, "3D is here: Point cloud library (PCL)," in *Proc. IEEE Int. Conf. Robot. Autom.*, May 2011, pp. 1–4.

[43] J. M. Coughlan and A. L. Yuille, "Manhattan world: Compass direction from a single image by Bayesian inference," in *Proc. 7th IEEE Int. Conf. Comput. Vis.*, vol. 2, Oct. 1999, pp. 941–947.

[44] 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," *Comput. Vis. Image Understand.*, vol. 154, pp. 192–205, Jan. 2017.

[45] Open Robotics. *Gazebo*. Accessed: Jan. 24, 2024. [Online]. Available: https://gazebosim.org/

[46] *Sweet Home 3D*. Accessed: Jan. 24, 2024. [Online]. Available: https://sweethome3d.com/

[47] A. Pérez Fornos, J. Sommerhalder, L. da Cruz, J. A. Sahel, S. Mohand-Said, F. Hafezi, and M. Pelizzone, "Temporal properties of visual perception on electrical stimulation of the retina," *Investigative Opthalmol. Vis. Sci.*, vol. 53, no. 6, p. 2720, May 2012.

[48] A. Perez-Yus, G. Lopez-Nicolas, and J. J. Guerrero, "Detection and modelling of staircases using a wearable depth sensor," in *Proc. ECCV Workshops III*, vol. 8927, no. 3, 2015, pp. 449–463.

[49] R. Mur-Artal and J. D. Tardós, "ORB-SLAM2: An open-source SLAM system for monocular, stereo, and RGB-D cameras," *IEEE Trans. Robot.*, vol. 33, no. 5, pp. 1255–1262, Oct. 2017.

[50] R. N. Kandalan and K. Namuduri, "Techniques for constructing indoor navigation systems for the visually impaired: A review," *IEEE Trans. Human-Mach. Syst.*, vol. 50, no. 6, pp. 492–506, Dec. 2020.

[51] D. Fox, W. Burgard, and S. Thrun, "The dynamic window approach to collision avoidance," *IEEE Robot. Autom. Mag.*, vol. 4, no. 1, pp. 23–33, Mar. 1997.

[52] V. V. Meshram, K. Patil, V. A. Meshram, and F. C. Shu, "An astute assistive device for mobility and object recognition for visually impaired people," *IEEE Trans. Human-Mach. Syst.*, vol. 49, no. 5, pp. 449–460, Oct. 2019.

[53] K. Patil, Q. Jawadwala, and F. C. Shu, "Design and construction of electronic aid for visually impaired people," *IEEE Trans. Human-Mach. Syst.*, vol. 48, no. 2, pp. 172–182, Apr. 2018.

[54] F. Yu and V. Koltun, "Multi-scale context aggregation by dilated convolutions," in *Proc. Int. Conf. Learn. Represent. (ICLR)*, 2016, pp. 1–13.

[55] L.-C. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille, "DeepLab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected CRFs," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 40, no. 4, pp. 834–848, Apr. 2018.

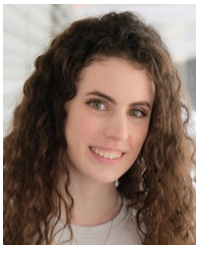
[56] B. Zhou, H. Zhao, X. Puig, S. Fidler, A. Barriuso, and A. Torralba, "Scene parsing through ADE20K dataset," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 5122–5130.

[57] T. Xiao, Y. Liu, B. Zhou, Y. Jiang, and J. Sun, "Unified perceptual parsing for scene understanding," in *Proc. Eur. Conf. Comput. Vis.*, Sep. 2018, pp. 418–434.

[58] B. Zhou, H. Zhao, X. Puig, T. Xiao, S. Fidler, A. Barriuso, and A. Torralba, "Semantic understanding of scenes through the ADE20K dataset," *Int. J. Comput. Vis.*, vol. 127, no. 3, pp. 302–321, Mar. 2019.

[59] D. Gutierrez-Gomez, W. Mayol-Cuevas, and J. J. Guerrero, "Inverse depth for accurate photometric and geometric error minimisation in RGB-D dense visual odometry," in *IEEE Int. Conf. Robot. Autom. (ICRA)*, Sep. 2015, pp. 83–89.
[60] D. Avraham, J.-H. Jung, Y. Yitzhaky, and E. Peli, "Retinal prosthetic vision simulation: Temporal aspects," *J. Neural Eng.*, vol. 18, no. 4, Aug. 2021, Art. no. 0460d9.
[61] M. S. Humayun, J. D. Weiland, G. Y. Fujii, R. Greenberg, R. Williamson, J. Little, B. Mech, V. Cimmarusti, G. Van Boemel, G. Dagnelie, and E. de Juan, "Visual perception in a blind subject with a chronic micro-electronic retinal prosthesis," *Vis. Res.*, vol. 43, no. 24, pp. 2573–2581, Nov. 2003.
[62] G. S. Brindley and W. S. Lewin, "The sensations produced by electrical stimulation of the visual cortex," *J. Physiol.*, vol. 196, no. 2, pp. 479–493, May 1968.
[63] J. F. Rizzo, J. Wyatt, J. Loewenstein, S. Kelly, and D. Shire, "Perceptual efficacy of electrical stimulation of human retina with a microelectrode array during short-term surgical trials," *Investigative Opthalmol. Vis. Sci.*, vol. 44, no. 12, p. 5362, Dec. 2003.
[64] J. S. Hayes, V. T. Yin, D. Piyathaisere, J. D. Weiland, M. S. Humayun, and G. Dagnelie, "Visually guided performance of simple tasks using simulated prosthetic vision," *Artif. Organs*, vol. 27, no. 11, pp. 1016–1028, Nov. 2003.
[65] N. R. Srivastava, "Simulations of cortical prosthetic vision," in *Visual Prosthetics*. Cham, Switzerland: Springer, 2011, pp. 355–365.
[66] W. H. Li, *A Fast and Flexible Computer Vision System for Implanted Visual Prostheses*. Cham, Switzerland: Springer, 2015, pp. 686–701.
[67] S. C. Chen, N. H. Lovell, and G. J. Suaning, "Effect on prosthetic vision visual acuity by filtering schemes, filter cut-off frequency and phosphene matrix: A virtual reality simulation," in *Proc. 26th Annu. Int. Conf. IEEE Eng. Med. Biol. Soc.*, Sep. 2004, pp. 4201–4204.

ALEJANDRO PEREZ-YUS received the Ph.D. degree in systems engineering and computer science from the University of Zaragoza, Spain, in 2018. He is currently an Assistant Professor with the Department of Computer Science and Systems Engineering, University of Zaragoza. He is a member of the Robotics, Computer Vision and Artificial Intelligence Group, and the Aragon Institute of Engineering Research (I3A). His current research interests include computer vision, artificial intelligence, and robotics, particularly in topics, such as scene understanding, unconventional cameras (e.g., omnidirectional and depth cameras), and the application of computer vision and robotic techniques to assistive devices.



MARIA SANTOS-VILLAFRANCA received the bachelor's degree in industrial technologies engineering and the master's degree in industrial engineering from the University of Zaragoza, Spain, where she is currently pursuing the Ph.D. degree in systems engineering and computer science. She is also a member of the Robotics, Computer Vision and Artificial Intelligence Group, and the Aragon Institute of Engineering Research (I3A). Her current research interests include computer vision, deep learning, and the application of computer vision techniques to robotics and assistive devices.

JULIA TOMAS-BARBA received the bachelor's degree in industrial technologies engineering and the master's degree in industrial engineering from the University of Zaragoza, Spain, in 2021 and 2023, respectively, where she is currently pursuing the Ph.D. degree in systems engineering and computer science. She is also a member of the Robotics, Computer Vision and Artificial Intelligence Group. Her current research interests include computer vision and its application to assistive devices.

JESUS BERMUDEZ-CAMEO received the Ph.D. degree from the University of Zaragoza, Spain, in 2016. He is currently an Associate Professor with the Department of Computer Science and Systems Engineering, University of Zaragoza. He is also a member of the Robotics, Computer Vision and Artificial Intelligence Group, and the Aragon Institute of Engineering Research (I3A). His current research interests include computer vision, robotics and artificial intelligence, particularly in omnidirectional vision, 3D visual perception, geometric deep learning, and the application of computer vision and robotics techniques to assistive devices.

LORENZO MONTANO-OLIVAN received the bachelor's degree in electronics and automatic engineering and the master's degree in biomedical engineering from the University of Zaragoza, Spain, in 2016 and 2019, respectively. He is currently a member of Robotics Group, ITA. His current research interests include robotics, mapping, localization, navigation, and computer vision.

GONZALO LOPEZ-NICOLAS (Senior Member, IEEE) received the Ph.D. degree in systems engineering and computer science from the University of Zaragoza, Spain, in 2008. He is currently a Full Professor with the Department of Computer Science and Systems Engineering, University of Zaragoza. He is also a member of the Robotics, Computer Vision and Artificial Intelligence Group, and the Aragon Institute of Engineering Research (I3A). His current research interests include visual control, autonomous robot navigation, multirobot systems, and the application of computer vision techniques to robotics and assistive devices.

JOSE J. GUERRERO received the Ph.D. degree from the University of Zaragoza, in 1996. He is currently a Full Professor with the Department of Computer Science and Systems Engineering, University of Zaragoza. He is also a member of the Robotics, Computer Vision and Artificial Intelligence Group, and the Aragon Institute of Engineering Research (I3A). His current research interests include computer vision, particularly in 3D visual perception, robotics, omnidirectional vision, vision-based navigation, and the application of computer vision and robotics techniques to assistive devices.