Vision Processing for Assistive Vision: A Deep Reinforcement Learning Approach

Jack White, *Graduate Member, IEEE*, Tatiana Kameneva, *Member, IEEE*, and Chris McCarthy, *Member, IEEE*

***Abstract*—There is increasing interest in using computer vision and machine learning to enhance human decision making with computer-mediated assistive vision systems. In particular, retinal implants are a rapidly advancing technology offering individuals suffering vision loss due to retinal dystrophies, an opportunity to restore partial vision. However, the visual representations achievable with current and near-term implants are severely limited in resolution and contrast, placing high importance on the selection of visual features to convey via the implant. Using vision processing algorithms on camera-captured input, functional outcomes can be enhanced with such devices. To this end, we propose a novel end-to-end vision processing pipeline for prosthetic vision that learns task-salient visual filters in simulation offline via deep reinforcement learning (DRL). Once learnt, these filters are deployable on a prosthetic vision device to process camera-captured images and produce task-guiding scene representations in real-time. We show how a set of learnt visual features enabling a virtual agent to optimally perform the task of navigation in a 3-D environment can be extracted and applied to enhance the same features in real world images. We evaluate and validate our proposed approach quantitatively and qualitatively using simulated prosthetic vision. To our knowledge, this is the first application of DRL to the derivation of scene representations for human-centric computer-mediated displays such as prosthetic vision devices.**

***Index Terms*—Assistive vision, computer vision, deep learning (DL), image processing, prosthetic vision, reinforcement learning, vision processing, visual prosthesis.**

I. Introduction

NEURAL degenerative diseases such as retinitis pigmentosa (RP) and age-related macular degeneration (AMD) cause loss of visual function through the deterioration of retinal cells. However, connection with the visual cortex is preserved by surviving retinal ganglion cells, allowing electrical stimulation delivered by a surgically implanted electrode array to restore partial vision in the form of artificially generated light spots. These percepts, called phosphenes, are typically described as "points of light" in the visual field of the implant recipient. Most retinal prostheses receive input from an externally worn camera, however, notable exceptions such as the subretinal Alpha-AMS implant capture light via an eye-resident photodiode array [1], [2].

Despite recent clinical trial successes, current and near-term retinal implants remain severely limited in the functional outcomes they can support. This has motivated consideration of vision processing techniques for selectively extracting and enhancing salient visual information to facilitate basic visual competencies, and guide actions. Examples include contrast or edge enhancements, structural cues [3], or surface proximity [4]. Phosphene visualizations are typically hand-crafted to encode specific features based on an underlying intuition that specific visual cues will lift functional performance. While patient and simulated prosthetic vision trials have indeed demonstrated functional outcomes can be improved using such methods [5], [6]. These results are typically focused on tasks with a single specific need for visual guidance such as object localization, orientation-to-target, or collision avoidance. However, there remains a significant gulf between the current functional performance achieved with retinal prostheses and the needs of vision-guided activities in real world settings.

An alternative approach is to *learn* visual representations for prosthetic vision. Deep learning, and in particular recent advances in deep reinforcement learning (DRL) offer the as yet unexplored possibility of training deep neural networks (DNNs) for prosthetic vision within the offline simulation of high priority tasks a prosthetic vision system aims to support. By training a virtual agent to learn to perform vision-based tasks within a simulated 3-D environment, a set of visual features can be extracted from the convolutional layers of the underlying DNN that facilitate the agent's ability to optimally perform the task. The visual features learnt are thus, inherently task-based, encoding visual cues inferred to be of highest relevance to the task being performed, and the current environmental context. This gives rise to the possibility of generating context-aware vision processing filters for prosthetic vision via offline learning in simulation [7].

To this end, we contribute a novel end-to-end vision processing pipeline that directly addresses the need for task-based, context-aware scene visualizations for human-centric computer-mediated displays such as prosthetic vision. Our learning-based vision processing model, as summarized in Fig. 1(b), exploits pretraining of a software agent on specifically designed simulated tasks in order to learn and extract task-salient visual features. From this offline learning process, image filters are learnt that may be deployed on the vision processing unit of

Manuscript received May 11, 2020; revised May 28, 2021; accepted September 12, 2021. Date of publication November 24, 2021; date of current version January 14, 2022. This article was recommended by Associate Editor Hantao Liu. *(Corresponding author: Jack White.)*

The authors are with the Department of Computer Science and Software Engineering, Swinburne University of Technology, Hawthorn, VIC 3122, Australia (e-mail: jawhite@swin.edu.au; tkameneva@swin.edu.au; cdmccarthy@swin.edu.au).

This article has supplementary material provided by the authors and color versions of one or more figures available at https://doi.org/10.1109/THMS.2021.3121661.

Digital Object Identifier 10.1109/THMS.2021.3121661

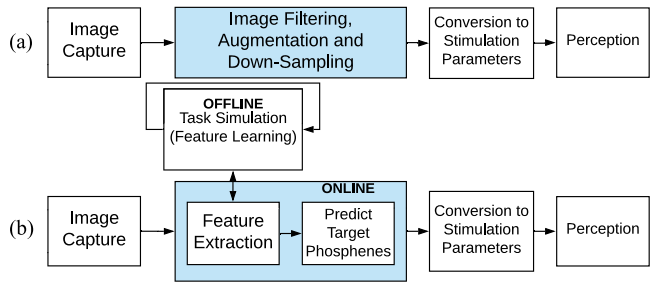

Fig. 1. Vision processing for prosthetic vision, showing (A) the traditional hand-crafted approach that models a particular visual representation and selects features to visualize that model. (B) Our proposed DRL-based pipeline, where feature selection and filtering is learnt via a simulation of a task offline. The learnt feature maps can then deployed online as filters posttraining to produce target phosphene visualizations.

a retinal prosthesis, allowing camera-captured images to be enhanced using the same learnt filters. Focusing on orientation and mobility (OM) as a case study, we present examples of our model's potential to encode visual features of importance to navigation in structured environments and discuss how our model is readily extendable to other visual activities of daily living. From this, we argue that this general model for human-centric vision processing offers key advantages for both the development of task-targeted and customisable vision processing pipelines, and its scalable deployment on prosthetic vision devices. Furthermore, this methodology can be extended to other modalities in assistive vision including tongue-based electrotactile displays [8] and auditory devices [9].

The rest of article is structured as follows. Section II presents background and literature relevant to the key contributions of the article. Section III describes our proposed vision processing pipeline, providing intuition on the generality of the approach to broader vision-based tasks and environments. Section IV outlines the application of the pipeline to an OM task. Section V presents the results of our case study establishing the learnt visual representation on simulated and real images. Section VI discusses our results, highlights implications, as well as open questions associated with our vision processing model. Finally, Section VII concludes this article.

II. Background and Related Work

A. Vision Processing for Prosthetic Vision

In general, vision processing for prosthetic vision is the modification of captured photosensor data to produce a functional representation for a given prosthetic vision device. We consider the vision processing model to be a mapping from a high-resolution image ψ with high-dynamic range captured at a high-frequency to a low-resolution visualization ϕ at low-dynamic range on a low-frequency stimulation device such as a retinal implant [10]

$$\phi = f(\psi). \tag{1}$$

As depicted in Fig. 1(a), the traditional pipeline for vision processing is to augment captured images to show visualizations emphasising hand-crafted features that facilitate basic task performance.

Both clinical and simulated prosthetic vision trials have established the effectiveness of vision processing for improving functional performance with retinal implants. Recent clinical trials with a 24 channel suprachoroidal implant have shown the application of the Lanczos2 reconstruction filter when subsampling input images can significantly improve functional performance on a light localization task when compared with single-point sampling [5], [11]. Encoding high-resolution images in a depth-based representation has also been shown in simulated and patient trials [12], to improve performance on a mobility and object avoidance task [13]. Such methods have shown promise in enhancing critical visual information for retinal prosthesis recipients and have established the vital role of vision processing in the prosthetic vision pipeline.

While such methods are capable of mapping visual input to action cuing, DRL architectures provide a framework that more directly addresses the aim of learning visual features for action guidance. DRL uses a conventional DNN structure with convolutional layers for feature extraction from image-based inputs. However, DRL algorithms also incorporate a vision-based task into its optimization procedure, meaning that features are learnt with respect to the needs of the task, and thus, always task-relevant.

B. Deep Reinforcement Learning

We propose DRL as a novel feature extraction method for vision processing. Thus, we provide a brief overview of reinforcement learning here. For a comprehensive review of DRL methods (see [14]).

DRL trains a virtual agent interacting with a simulated environment ϵ to complete a given task in discrete time-steps t. By taking an action a_t in response to a sampled image of the agent's immediate environment called a *state* s_t, the agent makes progress toward completing its task. Upon taking action a_t according to an action policy π, the agent receives a numerical reward value r as feedback on its performance and transitions into a following state s_{t+1}. The policy π is a mapping from input states to actions $(A \rightarrow S)$ that dictates action selection, and thus, can be considered a representation of the agent's decision-making strategy.

The agent's goal is to learn an effective policy that guides its actions to achieve high performance on its task through which r provides feedback. Rewards can be received as either positive or negative, conditional on the benefit that the action taken had toward achieving high-performance on the given task. The return on rewards R is the cumulative reward at time t defined as

$$R_t = \sum_{k=0}^{T} \gamma^k r_{t+k}. \tag{2}$$

This quantity represents the overall reward received in a given time period. A discount factor $\gamma \in (0, 1]$ is included to ensure the convergence of the return. The terminal state s_T is received in an episodic environment when the agent completes its task or is artificially instructed to stop. An episode is defined over the time period $t_0 : T$.

In order to learn an action-selection strategy, the agent learns to estimate the future reward it will receive based on its current state and selected action. We define the estimation of the value of a state s_t under a policy π as

$$V_\pi(s) = E\left[R_t | s_t = s\right] \tag{3}$$

where $V_\pi(s)$ is the state-value function. This quantity is the estimated return R_t that the agent will receive when acting under the policy π in the state s_t. We further define a second quantity, the state-action value function $Q_\pi(s, a)$, as the expected return on reward for selecting an action a_t in state s_t under the policy π as

$$Q_\pi(s, a) = E\left[R_t | s_t = s, a\right]. \tag{4}$$

To improve the performance of the agent, the expected return $E[R_t]$ is optimized to learn a parameterized policy function $\pi(s|a;\theta)$. Reinforcement learning algorithms that have an optimization framework that learns a policy are called policy gradient methods. By optimizing the policy, we are permitting the agent to learn an optimal mapping of states to actions that in turn ensures peak returns on rewards for any given state and action.

The role of the DNN is informally to assess the benefit of the agent's current situation and predict the future rewards it will obtain. If the DNN can successfully predict the future rewards the agent will obtain based on an image of its immediate environment, then the appropriate set of actions it needs to select to achieve the highest possible reward is trivial. Therefore, the DNN is learning a mapping from input images of the agent's state to the appropriate action that is dictated by the structural features in the image that the agent learns to focus on during training. We refer to the DNN's cohesive "knowledge" of the environment and task as its *representation*.

III. Task-Based Vision Processing

A. Model

Our pipeline for vision processing consists of two stages; offline feature learning and visualization through DRL, and the online deployment of the learnt filters for real-time vision processing. Fig. 1 presents our proposed model (B) and the traditional vision processing model (A). While the traditional approach generates a visual representation based on hand-crafted enhancements (e.g., edge mapping), our proposed model learns these features via offline simulation, which may be downloaded to a device as neural network weights. Specifically, a virtual agent is trained offline to learn to automatically extract task-salient features from its environment and encode them as filters in a DNN. The trained filters in the convolutional layers of the DNN perform the image filtering on captured images, emphasising the salient characteristics of the image. Captured image frames can be converted into a low-dimensional representation on the basis of the learnt filters and converted into stimulation parameters for visual perception.

B. Feature Learning for Visuo-Motor Tasks

The DRL-based framework is made up of two customisable features; the task model and the virtual environment. In previous works, agent's have been trained on both 2-D and 3-D environments on tasks such as locomotion [15], complex movement mimicking [16], and gameplay [17], [18]. A given virtual task is defined by the reward system that the agent uses as feedback and the permissible actions it can perform within the environment. Configuring this task setting within an environment that provides context for the task, permits the agent to learn an optimal action-selection policy that is calibrated for the customized task and environment. Thus, the framework is configurable for any number of potential tasks that are critical for functional outcomes for implant recipients.

Our vision processing pipeline is applicable to any common vision-guided tasks, including reach-and-grasp and object tracking. In this article, we consider the case study of OM. To model this task in simulation, we construct a reward system to promote optimal execution of the task. Modeling the task in this way outlines for the agent, as a proxy for the implant recipient, the constraints of the task being undertaken, dictating the type of image-to-action policy it will learn and by association, the underpinning visual features. Our model assigns a large reward for reaching a desired landmark and smaller rewards throughout the environment to encourage exploration. The virtual environment the agent acts within additionally affects the policy it learns. To encode a navigation task indoors, we model the environment to contain similar structural characteristics to indoor environments (i.e., symmetric edges, walls, etc.). The learnt features may then generate heightened responses to characteristics of the environment that facilitate the agent's ability to navigate.

Finally, we produce target phosphene visualizations from input images on the basis of the learnt filters as a proxy for visual perception of a person with an implant. As previously mentioned, a useful visualization is to directly filter images via the convolutional filters to highlight salient information in input images. In this way, we can directly convert task-salient information into an interpretable visual perception. This step finalizes the vision processing of this pipeline; taking an image and converting it into a target phosphene representation by training a single DNN.

IV. Case Study: Vision Processing for OM

In this section, we demonstrate the application of the proposed DRL-based model to the production of a vision processing pipeline to support basic OM. We choose OM as it represents a critical, high-priority capability to be restored with current and near-term using prosthetic vision devices [19].

A. Task-Based Feature Learning for OM

1) Simulated Environment: A critical outcome for prosthetic vision research is to restore a level of visual perception that is capable of supporting navigation in typical domestic and occupational environments. Therefore, modeling an environment that captures generic environmental features is motivated by

the need to provide visual cues that generalize to a range of real-world environments and tasks. Furthermore, preserving a 3-D representation that maintains the height and movement of a human perspective is important to maximize the transferability of the learnt visualization to human users.

To this end, we selected the 3-D DeepMind Lab maze environment "NavMazeStatic01" that possesses a task setting designed for basic exploratory navigation [20]. The objective is to reach a landmark within a virtual maze that randomly spawns the agent at the beginning of each episode or when the agent arrives at the landmark. For reaching the landmark, the agent receives a large, numerical, positive reward but additionally, smaller numerical rewards are scattered within the maze to emphasize exploration. Ensuring the agent is prioritising exploration is important for the agent to learn a general policy for action-selection rather than overfitting to its environment. The environment is thus well-suited for learning visual features and cues that will support basic OM in a typical indoor environment given the structural symmetry and landmark-based reward system.

For this OM case study, the agent was restricted to six actions; move forward, backward, left and right, look left and look right. Movement velocity was fixed in all directions and horizontal rotations (looking left and right) were quantized to 20° motions. These restrictions were necessary as a trade-off for ease of training the DNN while using a relatively complex 3-D environment. Furthermore, the environment additionally featured several limitations including no negative rewards (i.e., penalties) for collisions, no self-moving objects and a singular major landmark to reach. Note, however, that such additional conditions can be introduced, after which the model need only be retrained again.

2) Training: To train our DRL agent, we used an implementation of the A3C algorithm [21]. Our architecture contains two convolutional layers and two recurrent neural network layers with long short-term memory (LSTM) cells as can be seen in Fig. 2. Each of these layers are essential for the agent's training and resultant trained filters. The convolutional layers enable localized feature extraction and encoding into visual filters while the recurrent layers provide time-varying decision making through the memorization of short term interactions over multiple frames. This network structure follows the methods of the Nav A3C model that supports depth learning as a part of the architecture [22]. In addition, we selected Nav A3C given its efficacy in learning a 3-D environment with a navigation task setting. Maintaining depth learning as a part of the representation it learns enables a more robust policy for navigation, further permitting the full use of RGB and RGB-D image data as input to the network. We present the outcome of our training for the Nav A3C algorithm in Fig. 3.

Nav A3C learns low-dimensional encodings of higher resolution images as filters in the convolutional layers of the DNN. In our implementation, input images from the environment were sampled as 84×84 pixel RGB-D images but converted to grayscale for the purpose of training to meet the standards of current retinal implant devices. The visual features are encoded into 16 convolutional filters of dimensions 8×8 and 32×16 filters of 4×4 in the first and second convolutional layers,

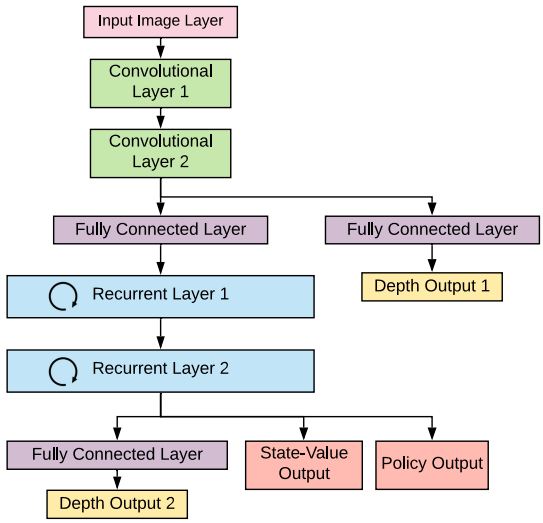

Fig. 2. Diagrammatic representation of the Nav A3C DNN architecture. The filters we use to filter captured images are learnt in the convolutional layers (Green) where images are encoded into a low-dimensional representation of the environment. The recurrent layers (Blue) that use LSTM cells afford the agent the ability to memorize sequenced interactions over many images, permitting it to exhibit complex behaviour like path planning.

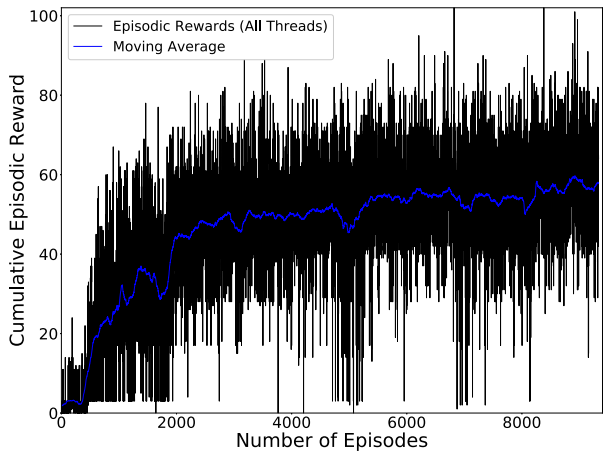

Fig. 3. Time series plot showing the increase in cumulative reward (rewards received over an entire episode) during training for all asynchronous threads. Images of the agent's FOV were sampled every 4 frames thus, a set of new 900 images were passed to the network per episode. A reward of 10 is allocated for reaching the goal within the maze, showing that in later stages of training the goal is reached multiple times per episode.

respectively, as demonstrated in Fig. 10. From the input RGB-D images, a 4×16 depth map was sampled from which network gradients were calculated from a comparison to a predicted depth map. While inclusion of a depth channel improves training performance, it is not a necessary component of our proposed vision processing model.

B. Validation of Training Methods

1) Saliency Mapping: In the context of vision processing for prosthetic vision, it is critical that learnt filters are validated before deployment with patients. For this purpose, we adapted the visualization technique of [23] to generate saliency maps

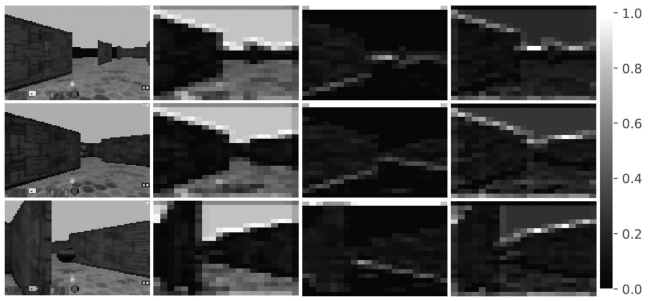

Fig. 4. Filtered images sampled from the simulated environment showing in columns from left-to-right: The original image, followed by the three selected feature maps. The strength of the filter response is indicated by the adjacent normalized heat scale.

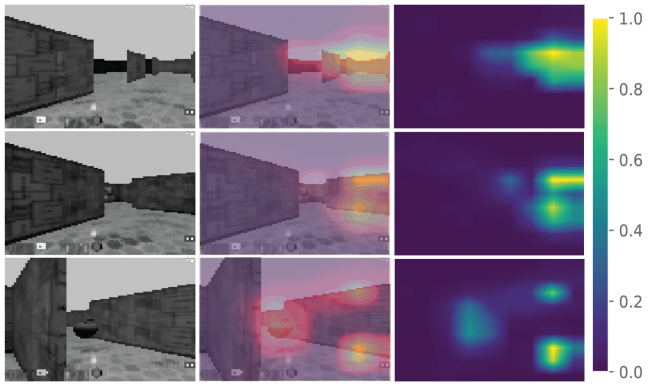

Fig. 5. Saliency maps of the agent's perceived importance of environmental features. (First column): Raw grayscale images of the environment. (Second column): Overlays of saliency maps onto raw environmental images that highlight the most important regions for action selection. (Third column): Raw saliency maps where regions of higher contrast indicate a high level of importance for the agent to select an action.

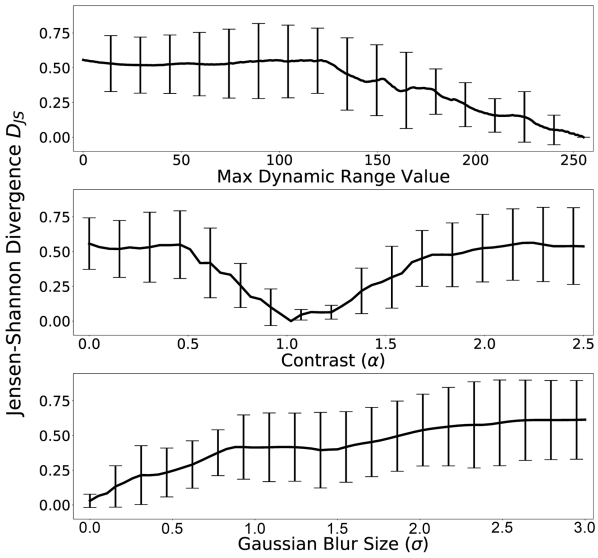

Fig. 6. Three plots showing the sensitivity the trained DNN has to noisy augmentations such as dynamic range, contrast (α) and smoothing (σ). These plots validate that training within this environment causes the agent to be particularly sensitive to contrast given the sharp distribution around its lowest point where no augmentation is applied. Statistics for each plot were generated from a set of 10 images.

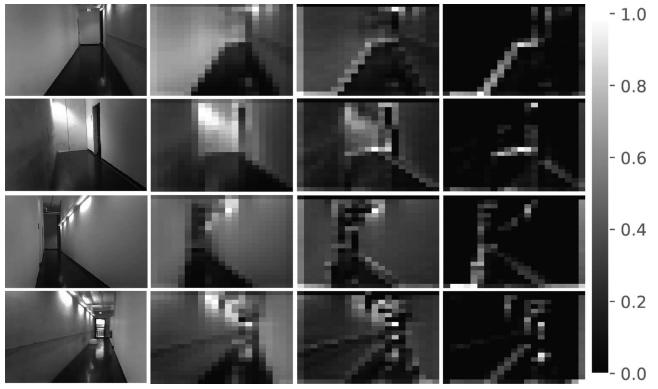

Fig. 7. Feature maps generated from the convolution of learnt filters over real input images. Strength of filter responses are relative to the provided heat scale. First column: Real images captured via RGB-D camera of simple and symmetric environments. Remaining columns depict resulting images after convolution of the same learnt filters depicted in Fig. 4. Notably, the same features are emphasizes as in Fig. 4 by each individual filter.

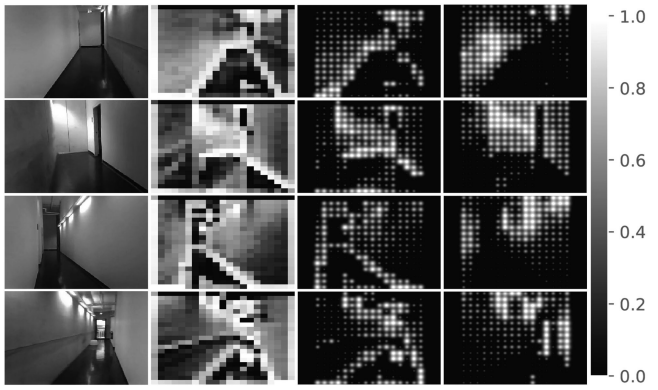

Fig. 8. Real-world image examples (first column) with corresponding combined feature maps (second column) and simulated phosphene image derived from this (third column). These phosphene images are compared to traditional intensity mapping in the (fourth column).

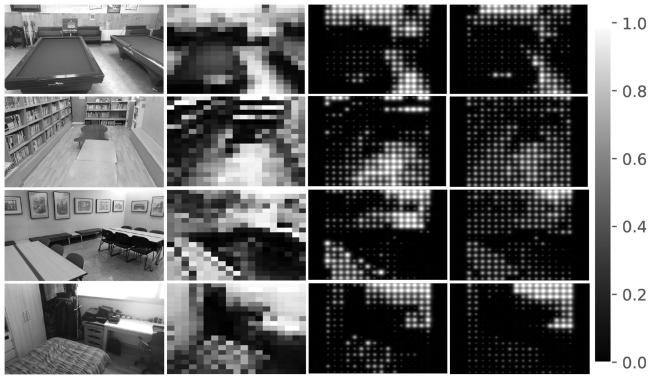

Fig. 9. A set of real images (first column) sampled from the DIML/CVLAB RGB-D dataset displaying variably cluttered environments with associated compound feature maps (second column) and target phosphene visualization (third column) [24]. We again, compare the results of the third column with an intensity-based representation in the (fourth column).

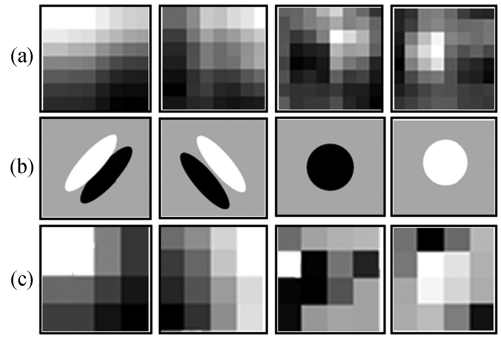

Fig. 10. Trained filters from the two convolutional layers of the DNN. The larger filters (A) are derived from the first layer with kernel size (8 x 8) and the smaller filters (C) are a sampled set of filters from the second layer with kernel size (4 x 4). Each set of filters were smoothed with a mean spatial filter to provide a more intuitive representation of the learnt features. (B) Receptive fields of visual neurons, from the top: ON-center cell, OFF-center cell, two directionally selective cells.

highlighting aspects of input images that the agent perceived as being salient to task completion. The technique exploits the agent's policy $\pi(s, a)$ estimation process, which makes a softmax estimation of the action the agent will take for a particular input. Given an image of the agent's current state s_t, the network will estimate an action a_t to take. By occluding the input image using a Gaussian blur, the output state-value will consequently be changed. Therefore, two differing policy estimations are made; one for the original image I_t and the other for the occluded image I'_t. Taking the difference between these values yields a saliency score $S_\pi(t, i, j)$ for the pixel region with coordinates (i, j)

$$S_\pi(t,i,j) = \frac{1}{2}||\pi_u\left(I_{1:t}\right) - \pi_u\left(I'_{1:t}\right)|| \tag{5}$$

where I' is the occluded pixel region of the input image if $k = t$ and an unperturbed region otherwise. The term $\pi_u(I_{1:t})$ defines the softmax prediction of the policy output. Sequentially blurring separate regions over the whole image, calculates a saliency score for each pixel region, indicating the significance that feature holds with respect to the policy output. This creates a saliency map that may be overlaid on the input image. We applied the original 2-D method to a 3-D environment to meet the practical needs of the study.

2) Training Sensitivity: We performed a sensitivity analysis on the trained DNN to glean the most important image characteristics for the agent to select an action. By analyzing the sensitivity of the network to key visual properties, specifically dynamic range, contrast and spatial frequency, we provide evidence toward their relative impact on the DNN's selection of features in the real images presented in Section V-D.

Similar to the above, we applied a noisy augmentation to the image and sequentially increased its severity, inputting the augmented image into the network each time to obtain a softmax estimation over the selectable actions. From these estimations, we calculated the Jensen-Shannon divergence D_{JS} that reflects the disparity between two probability distributions, where $D_{JS} = 0$ reflects two perfectly equal distributions and $D_{JS} = 1$ shows completely contrasting distributions

$$D_{JS} = \sqrt{\frac{D_{KL}(p||m) + D_{KL}(q||m)}{2}} \tag{6}$$

where D_{KL} is the Kullback–Leibler divergence, p and q are the two compared probability distributions and m is the pointwise mean. We obtained D_{JS} values from the softmax policy output comparing an occluded input and unoccluded baseline. A low-scoring D_{JS} value shows that the agent estimated the same distribution over its preferred actions and a high-scoring value represents a very different distribution over the selectable actions. Thus, the higher the D_{JS} the more the action selection deviates from its optimal choice. We chose three augmentations that impact three fundamental visual properties including contrast, dynamic range and edge smoothing. We argue that these augmentations should sufficiently demonstrate the relative importance of these visual properties for the DNN to select actions from visual input and its sensitivity to typical environmental noise. The contrast was augmented via the following:

$$g(i,j) = \alpha \cdot f(i,j) \tag{7}$$

where α is the gain parameter that augments the contrast of the input pixel located at (i, j). To implement edge smoothing, we applied a Gaussian kernel and varied the severity of blurring by the standard deviation σ. For constraining the dynamic range of an image, we mapped pixel values to a fixed number of gray levels. Training samples were encoded as 8-b, UTF-8 images and for this analysis, were constrained by reducing the possible gray levels from $255 \rightarrow 0$ in integer steps.

C. Target Phosphene Prediction

A critical step in the methodology is extracting the agent's learnt representation and conveying the result as an intuitive visualization for task guidance. We outline a possible method in this article but note that visualization of the filtered output may be achieved a number of ways which ultimately require human trials to assess.

As a possible approach, we add sets of N normalized images filtered by the trained convolutional filters. Formally, we generalize this process as

$$g(i,j) = \frac{1}{N}\sum_{k=1}^{N} p_k(i,j) \tag{8}$$

where p is a pixel located at (i, j) from the kth feature map. Numerous approaches may be adopted for the choice of feature maps to generate visualizations from. For visual clarity, we selected features based on their similarity to feature-selection simple cells in the primate visual cortex. The resulting hybrid feature map $g(i, j)$ forms the target visual representation for encoding via an implant-specific and patient-calibrated stimulator as phosphenes as per Fig. 1.

V. Results

We validated our vision processing pipeline using the techniques described in Sections IV-B and IV-B2 on both filtered

images sampled directly from the virtual training environment, and on real images. We additionally compared target phosphene representations from our system with a baseline intensity encoding, using a prosthetic vision simulator. We present these results below.

A. Simulated Environment Images

Fig. 4 shows sampled images from the simulated environment and filter responses achieved using selected learnt filters from the DNN. Filter responses can be seen to highlight key environmental structure commonly associated with navigation. For example, column 2 filter responses indicate higher salience for horizontal planes, whereas columns 3 and 4 emphasize edges between wall-floor boundary and the sky-wall edges in the input image, respectively. Note that Fig. 4 shows 3 of the 16 filters in the first convolutional layer, and thus, only represents a subset of the required features to enable the agent to exhibit basic OM skills for navigating to a landmark. Additionally, each feature map portrays a different feature, showing the scope and depth of the agent's understanding of basic navigation.

B. Saliency Maps

We apply the saliency technique described in Section IV-B on a set of simulated images. Fig. 5 displays the saliency maps generated for three simulated images. In deciding the optimal action for a given image, the agent will make a decision that encompasses features across the entire image. Despite this detail, there are clear regions that display higher task-saliency than others, indicating decisions on action selection are more heavily weighted toward these regions. As seen in Fig. 5, the peak regions of saliency correspond to visible regions the agent is planning on traversing toward. Since saliency is calculated on the basis of the policy prediction, these areas show the most important visual information for selecting an action.

To inspect the evolution of saliency over time, a video sequence[1] was also generated. In this video, the agent can be observed to dynamically alter its attention, placing focus on open areas where opportunity to move into new spaces occurs as well as on wall-floor and wall-sky interfaces for orientation, validating the agent's ability to recognize task salient information in each frame.

C. Sensitivity Analysis

Employing the methods described in Section IV-B2, we show the results of the network sensitivity study in Fig. 6, which establishes the importance of contrast for action estimation. Each plot was generated over a parameter range that was constrained at the point when the distributions began to plateau. This excludes the dynamic range that was assessed over all 256 total gray levels. The variance observable in Fig. 6 is reflective of the uncertainty in action predictions when an augmentation is applied. As the severity of an augmentation is increased, it can be seen that the uncertainty also monotonically increases.

Fig. 6 shows that compared to the dynamic range and Gaussian blur distributions, the contrast reaches its maximum value much more directly. This plot is a validation of the importance that contrast has in the agent's identification of salient features. Furthermore, it highlights a limitation of the 3-D environment used in this study. While contrast may have enabled the agent to easily discern features that facilitate its ability to navigate, it limits the generalization of feature recognition in real-world images. This consideration will be addressed in future research.

D. Real-World Images

To establish the real-world transferability of the learnt filters, real images depicting simple indoor structured scenes were captured using an Orbbec Persee RGB-D camera (Orbbec 3D Technology International Inc. MI USA), down-sampled to an 84×84 resolution, and converted to grayscale to meet the specifications of the DNN input layer, and filtered. Fig. 7 depicts shows samples showing outputs from the trained filters, and their ability to highlight task-salient information in each scene.

Fig. 7 results demonstrate the agent has recognized environmental features such as surface planes (column 2), edges at high-contrasting boundaries (column 3), and major edges (column 4). Similarities with Fig. 4 are also apparent, with the same features identified in both, demonstrating the transfer of learning from simulation to real scenes. This is further emphasized in supplementary feature map videos [2] that portray each feature map activating on a real sequence of input images.

Notably in Fig. 7, not all environmental features of the input images generate high filter responses. For example, in the second row, the left side wall-floor boundary does not generate a high response. As suggested by Fig. 6, this is likely a result of this region exhibiting lower contrast, as well as discrepancies between the camera's relatively free motion in the real-world scene (e.g., allowing camera rotation and tilt) compared with the highly planar motion simulated. This highlights a need to develop simulations that capture more realistic human motion (e.g., swaying during walking and head turning) to account for such discrepancies. However, it should be noted that other filter responses for the same frame make evident the totality of structural features to guide mobility down the corridor, emphasising the point that not all features need be evident in every frame to guide appropriate action choices over time.

E. Phosphene Visualization

On the basis of these feature maps, visualizations of target phosphene representations can be produced. These phosphene visualizations were generated using the simulation model described in [12], based on a theoretical 16×16 electrode array, with a dynamic range of 8 grey levels. These settings were selected based on current clinical results and near-term retinal implants currently under development. Importantly, we note that idealized phosphene simulations such as these do not accurately capture the true perceptual experience of current retinal implants. However, they do provide a useful abstraction of key

[1]https://jwhite-research.github.io/DRL-for-Assistive-Vision/

[2]https://jwhite-research.github.io/DRL-for-Assistive-Vision/

display constraints, providing a demonstration of the potential use of our pipeline for informing task-based visual guidance for OM.

As outlined in Section IV-C, we produced target phosphene visualizations using combined feature maps. Each feature map shown in Fig. 8 (column 2) was generated by adding together three normalized feature maps. Column 3 depicts the resultant phosphene encoding of these hybrid feature maps. The resultant visualization emphasizes structural features on real images, showing a clear enhancement of key surface boundaries. These visual elements of the displayed images highlight the importance of such features for basic navigation and orientation in simple scenes. An interesting consequence is the discernibility of particular structural features such as the doorway in the second row example of Fig. 8, and wall-floor shorelines.

In the final column of Fig. 8, we show examples of each image rendered in phosphenes as subsampled grayscale intensities from the original image, representing the traditional and dominant vision processing pipeline utilized in current retinal prostheses. When compared with our proposed technique, the advantages of the learnt feature enhancements are evident. While high-contrasting scenarios such as in the second row of Fig. 8 are adequately represented in both visualizations, it is in low contrast scenes that the structural enhancements of our approach offer advantages over intensity-based representations. In column 3 of Fig. 8, increased contrast is evident around important structural cues compared with the baseline intensity mapping (column 4), suggesting our approach may offer a better suited visualization for orientation and mobility.

To assess the agent's ability to recognize features in more structurally diverse scenes, we processed images from the DIML/CVLAB dataset that show a set of variably cluttered environments [24]. Fig. 9 compares the agent's behaviour to the baseline intensity-based representation. While comparative differences are less apparent than in Fig. 8, key differences can be observed. For example, the DRL representation increases contrast on objects of immediately importance to safe navigation such as in row 2, where the tables close to the observer are enhanced in the DRL-derived visualization (column 3) compared to the intensity-based image (column 4). Similarly, in row 4, the bed in the foreground exhibits enhanced contrast compared to the intensity-based rendering.

Critically, Fig. 9 demonstrates the stability of the trained filters when dealing with environments disparate from those used for training. In dealing with variably cluttered scenes, the filtered output approaches an intensity-like visualization in the absence of direct navigational cues. However, even in such cases, feature enhancements compared with intensity-based phosphene renderings are still apparent, including important objects such as the low contrast table in row 2, and foreground surface boundaries in row 3. While more subtle enhancements compared with Fig. 8, the use of just noticeable perceptual differences to mark such features in more cluttered settings arguably better serves the need to preserve dynamic range to convey more detail, as discussed in [25].

Finally, we provide image sequences from real-world scenarios in Fig. 11 to establish the efficacy of our pipeline in an evolving scene. Each real-world sequence was taken in an indoor, structurally simple environment with high-contrast. We show the stability of our image filters in an evolving scene by varying the height and perspective of the frames as the scene progresses. Despite being derived from a learning framework that has a locked visual perspective, the filters effectively identify salient information despite these variations. Compared to the intensity mapped phosphene visualizations, the learnt filters are more resilient to low-contrast and impart intuitive visualizations upon which safe action choices can be based. This is particularly noticeable in the second sequence, where large variations in the contrast occur. To emphasize this disparity, we provide a sequence of simulated images to indicate the features that are being displayed.

F. Learnt Feature Interpretability

The effectiveness of the learnt features to enhance functional outcomes for human viewers raises the question of how consistent these filters are with known human visual processing filters in the primary visual cortex. To this end, we assessed the features represented in the trained convolutional filters.

Results showed that most trained filters had a spatial structure, resembling ON and OFF subregions of receptive fields of neurons in the retina and V1 area of visual cortex. A neuron's receptive field is a region in a visual space where the light stimuli leads to changes in the neuron's response; the receptive field's spatial structure indicates which pattern of light and dark regions in the visual field are the most effective at driving the response. Schematics of biological receptive fields are illustrated in Fig. 10(b). Selected examples of trained filters are shown in Fig. 10(a) and (c). Inspection of these filters indicates similarities between biological receptive fields of visual neurons and our trained filters; the filters in the first two columns resemble directionally selective cells, the filters in the third column are similar to OFF-center cells, and the filters in the last column resemble ON-center receptive fields. We note that while the learned filters are intuitively interpretable, we are not trying to replicate the human visual system, rather the focus of our work is to develop a method for selecting the most important information that needs to be transmitted via a limited bandwidth. That these learnt filters resemble biological filters offers support for the interpretability and generality of the filtered signal.

VI. Discussion

Providing implant recipients with the most functionally effective perception possible is the goal of vision processing in prosthetic vision. Although the perception offered by state-of-the-art retinal implants is constrained by inherent biological drawbacks and the limitations of current neuro-stimulation technologies, vision processing can optimize the information transfer by intelligently determining and encoding high priority visual features. The method we present in this article represents an entirely new approach to vision processing for prosthetic vision, leveraging modern artificial intelligence techniques to determine visual representations embedded in the needs of task execution.

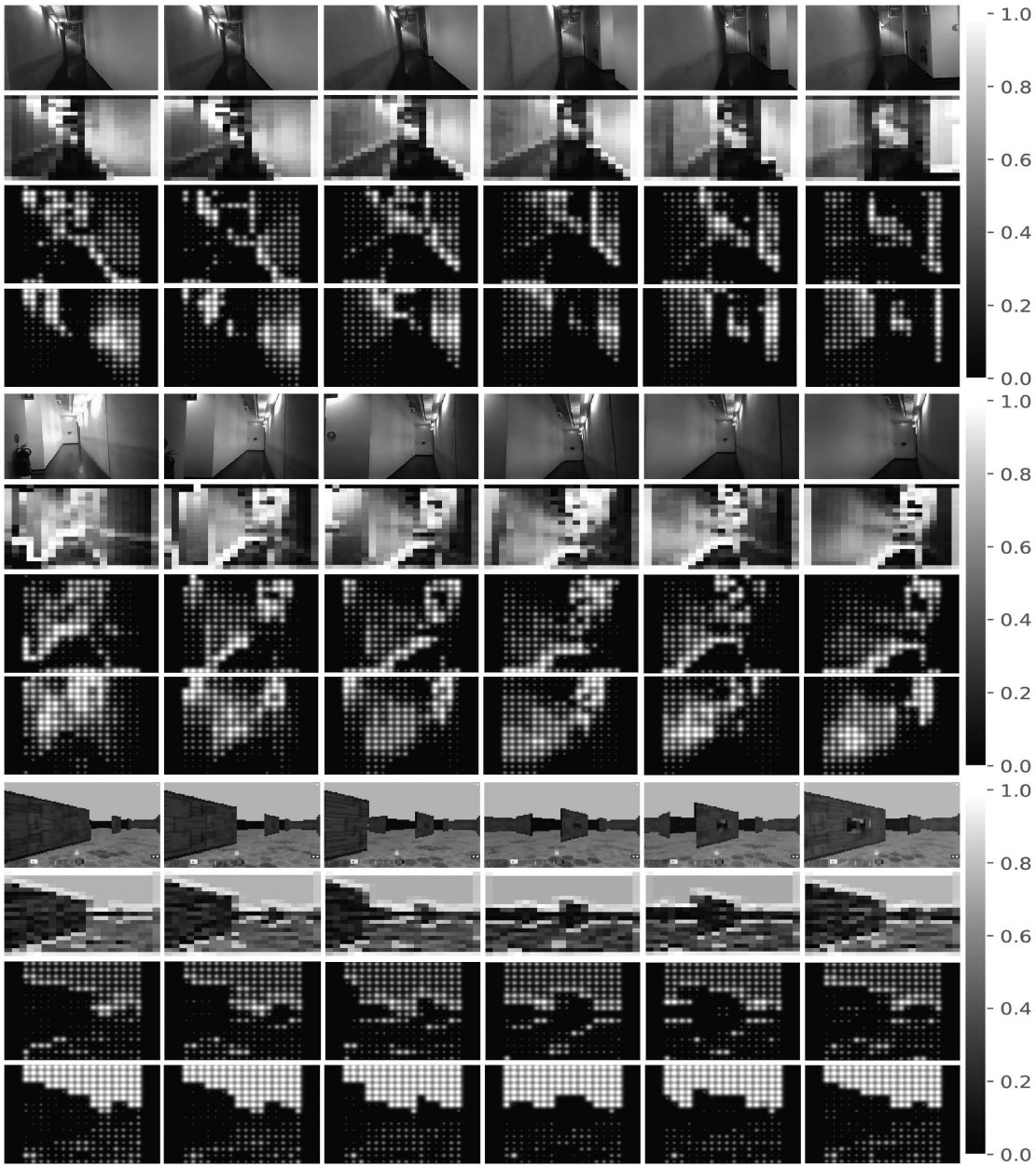

Fig. 11. Simulated and real image sequences (first row) with accompanying combined feature maps (second row) and target phosphene visualization (third row). We compare phosphene sequences generated from our combined feature maps with the standard intensity mapping approach (fourth row). Results demonstrate the stability of our technique over time and robustness in low-contrast scenarios. Filter responses are indicated by the adjacent colour bar.

The DRL-based approach defers all heavy computation to an offline process, allowing the learnt visual information to be deployed as real-time image filters on current implant systems, and can be swapped for other trained models to support different tasks as a set of network weights downloaded to the device. This means that image enhancements can be undertaken without the need to modify existing software, benefiting the rapid development and deployment of vision processing to support changing functional needs. Moreover, refinement of visual needs can be undertaken by modeling more specific environments for individual implant recipients (or cohorts), and training agents in these environments to undertake vision-based activities of daily living. The number of supportable tasks and environments is thus theoretically unlimited, albeit constrained by training-time requirements and fidelity levels achievable in the simulated task and environment.

Modeling an environment and agent in 3-D then constructing a reward system that captures the details of complex tasks is an ongoing area of research. The difficulty of training an agent scales with the complexity of the task and environment requiring custom DNNs with deeper networks to learn a robust policy. Training such networks requires a large amount of data in addition to computational time. A benefit of DRL on this front is the self-generated dataset it creates by sampling images from its environment. Furthermore, at the time of writing, several more robust DRL algorithms are available with more virtual

environment modeling software integrating deep learning directly into their interfaces [26], [27]. Future work will leverage these techniques as a means of modeling more complex environments and tasks, while learning more robust representations with cutting-edge DRL algorithms.

Currently, the stability of filtering real images can be improved and likely addressable by encoding more realistic physics into the environment such as varied textures and lighting effects. In the case study presented in Section IV, we also neglected factors such as collisions and stationary/moving obstructions as a means of simplifying the problem. Notably, encoding such elements into virtual environments is straight forward with modern software [28]. Additionally, applying transfer learning to our agent by training beyond the trained network presented here on alternative environments is a viable strategy for enhancing generalization and transferal from simulation to reality. Determining optimal methods for efficiently modeling complex environments is future work.

The proposed pipeline does not attempt to capture patient specific properties of stimulation, but rather, deliberately separates feature learning from the stimulation strategies that present these visualizations to patients. Such calibration of the visual representation to a specific patient could be performed once the base vision processing is deployed. This approach is consistent with other state-of-the-art vision processing pipelines which have been successfully trialled with patients. For example, [25] define a general augmentation method for prosthetic vision, combining captured images with estimated importance maps generated from any preferred method. Such frameworks, for which our approach may be easily integrated with, generates target scene visualizations which are subsequently down-sampled and encoded as phosphenes according to patient-calibrated stimulation parameters [29], [30]. To bridge the separation between visualization production and stimulation, future research may also consider integrating patients into the agent training process to better target patient-specific perceptual needs [31]. The goal of this research is to improve functional outcomes for implant recipients with task-specific visualizations. To verify that our model can improve performance, human testing in both simulated prosthetic vision settings and clinical trials will be required.

Importantly, the pipeline we propose is not limited to retinal prostheses themselves. More generally, the learning of visual features salient to human-centric task execution offers potential application to any computer-mediated visual guidance, including remote control of vehicles (i.e, drones, mining equipment, rescue robots, etc.), over bandwidth constrained display mediums such as augmented or virtual reality. Such examples represent analogous problems to what is described here, where optimal and timely decisions based on limited visual information is central to performance, and for which the proposed approach offers potential advantages.

VII. Conclusion

We have presented a novel, machine learning-based pipeline for vision processing in prosthetic vision. Harnessing the proven learning power of DNNs within the framework of DRL, we have presented a pipeline capable of learning task-based image filters in simulation, which may then be applied to real captured images. This represents a fundamentally new approach to vision processing for prosthetic vision. Results presented indicate that learning filters via task-based simulations in 3-D environments is a viable approach to detecting salient features in real-world scenes. We have demonstrated that the dynamically adapting model learnt by our trained agent can identify task-salient information that is contextualized by the environment and stage of the task it is undertaking. We further outlined that the proposed pipeline is adaptable to new tasks and environments. The pipeline was further identified as being readily usable in current implant devices given the agent is trained offline before deploying the visual features in an online setting. Our proposed technique offers a viable method for learning image filters that are task-relevant for mapping vision-to-action in future retinal implants and to our knowledge is the first approach in this field using applied DRL.

References

[1] K. Stingl *et al.*, "Interim results of a multicenter trial with the new electronic subretinal implant Alpha AMS in 15 patients blind from inherited retinal degenerations," *Front. Neurosci.*, vol. 11, 2017, Art. no. 445.

[2] K. Stingl *et al.*, "Artificial vision with wirelessly powered subretinal electronic implant alpha-IMS," *Proc. Roy. Soc. B: Biol. Sci.*, vol. 280, no. 1757, 2013, Art. no. 20130077.

[3] D. Feng and C. McCarthy, "Enhancing scene structure in prosthetic vision using ISO-disparity contour perturbance maps," in *Proc. 35th Annu. Int. Conf. IEEE Eng. Med. Biol. Soc.*, 2013, pp. 5283–5286.

[4] C. McCarthy and N. Barnes, "Time-to-contact maps for navigation with a low resolution visual prosthesis," in *Proc. Annu. Int. Conf. IEEE Eng. Med. Biol. Soc.*, 2012, pp. 2780–2783.

[5] N. Barnes *et al.*, "Vision function testing for a suprachoroidal retinal prosthesis: Effects of image filtering," *J. Neural Eng.*, vol. 13, no. 3, 2016, Art. no. 0 36013.

[6] W. L. D. Lui, D. Browne, L. Kleeman, T. Drummond, and W. H. Li, "Transformative reality: Improving bionic vision with robotic sensing," in *Proc. Annu. Int. Conf. IEEE Eng. Med. Biol. Soc.*, 2012, pp. 304–307.

[7] J. White, T. Kameneva, and C. McCarthy, "Deep reinforcement learning for task-based feature learning in prosthetic vision," in *Proc. 41st Annu. Int. Conf. IEEE Eng. Med. Biol. Soc.*, 2019, pp. 2809–2812.

[8] T. H. Nguyen, T. H. Nguyen, T. L. Le, T. T. H. Tran, N. Vuillerme, and T. P. Vuong, "A wearable assistive device for the blind using tongue-placed electrotactile display: Design and verification," in *Proc. Int. Conf. Control, Autom. Inf. Sci.*, 2013, pp. 42–47.

[9] J. P. Lerousseau, G. Arnold, and M. Auvray, "Visualizing sounds: Training-induced plasticity with a visual-to-auditory conversion device," 2021. [Online]. Available: https://doi.org/10.1101/2021.01.14.426668

[10] N. Barnes *et al.*, "The role of vision processing in prosthetic vision," in *Proc. Annu. Int. Conf. IEEE Eng. Med. Biol. Soc.*, 2012, pp. 308–311.

[11] N. Barnes *et al.*, "Vision processing with Lanczos2 improves low vision test results in implanted visual prosthetic patients," *Invest. Ophthalmol. Vis. Sci.*, vol. 55, no. 13, pp. 1802–1802, 2014.

[12] 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, 2014, Art. no. 0 16003.

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

[14] K. Arulkumaran, M. P. Deisenroth, M. Brundage, and A. A. Bharath, "Deep reinforcement learning: A brief survey," *IEEE Signal Process. Mag.*, vol. 34, no. 6, pp. 26–38, Nov. 2017.

[15] X. B. Peng, G. Berseth, K. Yin and M. Van De Panne, "DeepLoco: Dynamic locomotion skills using hierachical deep reinforcement learning," *ACM Trans. Graph.*, vol. 36, no. 4, pp. 1–13, Jul. 2020.

[16] X. B. Peng, P. Abbeel, S. Levine, and M. van de Panne, "DeepMimic: Example-guided deep reinforcement learning of physics-based character skills," *ACM Trans. Graph.*, vol. 37, no. 4, pp. 1–14, 2018.

[17] O. Vinyals *et al.*, "Grandmaster level in StarCraft II using multi-agent reinforcement learning," *Nature*, vol. 575, no. 7782, pp. 350–354, Nov. 2019.
[18] V. Mnih *et al.*, "Human-level control through deep reinforcement learning," *Nature*, vol. 518, no. 7540, pp. 529–533, 2015.
[19] A. Caspi *et al.*, "Combined eye-head vs. head-only scanning in a blind patient implanted with the Argus II retinal prosthesis," in *Proc. 8th Int. IEEE/EMBS Conf. Neural Eng.*, 2017, pp. 29–32.
[20] C. Beattie *et al.*, "Deepmind lab," GitHub.com, Accessed: Aug. 1, 2018. [Online]. Available: https://github.com/deepmind/lab
[21] M. Babaeizadeh, I. Frosio, S. Tyree, J. Clemons, and J. Kautz, "Reinforcement learning through asynchronous advantage actor-critic on a GPU," in *Proc. ICLR*, 2017.
[22] P. Mirowski *et al.*, "Learning to navigate in complex environments," 2017, *arXiv:1611.03673v3*.
[23] S. Greydanus, A. Koul, J. Dodge, and A. Fern, "Visualizing and understanding Atari agents," in *Proc. 35th Int. Conf. Mach. Learn.*, 2018, pp. 1792–1801.
[24] J. Cho, D. Min, Y. Kim, and K. Sohn, "Deep monocular depth estimation leveraging a large-scale outdoor stereo dataset," *Expert Syst. Appl.*, vol. 178, Sep. 2021, Art. no. 114877.
[25] C. McCarthy and N. Barnes, "Importance weighted image enhancement for prosthetic vision: An augmentation framework," in *Proc. IEEE Int. Symp. Mixed Augmented Reality*, 2014, pp. 45–51.
[26] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov, "Proximal policy optimization algorithms," 2017, *arXiv:1707.06347v2*.
[27] A. Juliani *et al.*, "Unity ML-agents toolkit," GitHub.com, Accessed: Nov. 5, 2021. [Online]. Avilable: https://github.com/Unity-Technologies/ml-agents
[28] E. Todorov, T. Erez, and Y. Tassa, "MuJoCo: A physics engine for model-based control," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2012, pp. 5026–5033.
[29] L. N. Ayton *et al.*, "A prototype suprachoroidal retinal prosthesis enables improvement in a tabletop object detection task," *Invest. Ophthalmol. Vis. Sci.*, vol. 56, no. 7, pp. 4782–4782, 2015.
[30] N. M. Barnes *et al.*, "Enhancing object contrast using augmented depth improves mobility in patients implanted with a retinal prosthesis," *Invest. Ophthalmol. Vis. Sci.*, vol. 56, no. 7, pp. 755–755, 2015.
[31] G. Li, R. Gomez, K. Nakamura, and B. He, "Human-centered reinforcement learning: A survey," *IEEE Trans. Human-Mach. Syst.*, vol. 49, no. 4, pp. 337–349, Aug. 2019.