Check for updates

Neurolight Alpha: Interfacing Computational Neural Models for Stimulus Modulation in Cortical Visual Neuroprostheses

Antonio Lozano[1], Juan Sebastián Suárez[2,3], Cristina Soto-Sánchez[2,3], Javier Garrigós[1](✉), Jose-Javier Martínez[1], José Manuel Ferrández Vicente[1], and Eduardo Fernández-Jover[2,3]

[1] Dpto. Electrónica, Tecnología de Computadoras y Proyectos, Universidad Politécnica de Cartagena, Cartagena, Spain
{amlo,javier.garrigos,jjavier.martinez,jm.ferrandez}@upct.es
[2] Instituto de Bioingeniería, Universidad Miguel Hernández, Alicante, Spain
{jsuarez,csoto,e.fernandez}@umh.es
[3] CIBER-BBN, Madrid, Spain
http://gruposinvestigacion.upct.es/grupos_ID/info_grupo.php?id=7,
http://bioingenieria.umh.es/,
https://www.ciber-bbn.es/

Abstract. Visual neuroprostheses that provide electrical stimulation along several sites of the human visual system constitute a potential tool for vision restoring for the blind. In the context of a NIH approved human clinical trials project (CORTIVIS), we now face the challenge of developing not only computationally powerful, but also flexible tools that allow us to generate useful knowledge in an efficient way. In this work, we address the development and implementation of computational models of different types of visual neurons and design a tool -Neurolight alpha- that allows interfacing these models with a visual neural prosthesis in order to create more naturalistic electrical stimulation patterns. We implement the complete pipeline, from obtaining a video stream to developing and deploying predictive models of retinal ganglion cell's encoding of visual inputs into the control of a cortical microstimulation device which will send electrical train pulses through an Utah Array to the neural tissue.

Keywords: Visual neuroprostheses · Neural encoding · Computational models · Artificial vision

1 Introduction

To restore the ability of the human neural system to function properly is one of the main purposes of neural engineering. In the context of this broad and multidisciplinary research field, where disciplines ranging from clinical neurology

© Springer Nature Switzerland AG 2019

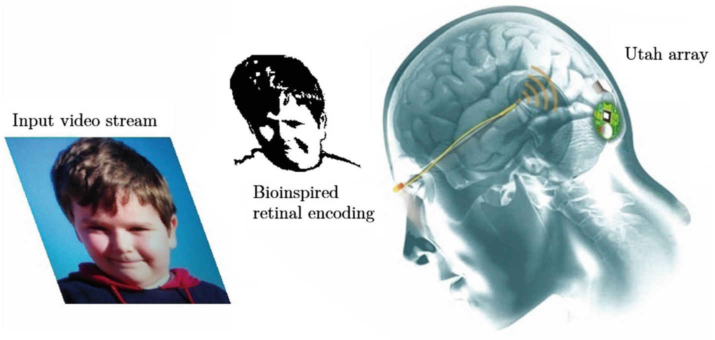

Fig. 1. A general idea of a cortical visual neuroprosthesis is composed of a camera obtaining a video stream, a encoding module and a stimulator which sends electrical pulses throughout an intracortical microelectrode interface.

to computational neuroscience, scientific advancement and engineering development has taken us until today's achievements: EEG-Based BCI [1], motor control BCIs with UTAH arrays [2,3], cochlear implants [5], retinal prosthesis [6], and Deep Brain Stimulation systems [7]. Regarding visual function recovering, several approaches are being extensively explored, such as optogenetics [8], biocompatible material design for neural interfaces [4] and neuromorphic computing for neuroprosthesis [9]. Specifically, several advances have been done in retinal prostheses, where several devices have been already clinically tested or are currently in use [10,11]. These devices are limited to a very specific causes of blindness, where the optic nerve function is intact. Cortical prosthesis appears as a potential solution to those blindness conditions for people with a functional visual cortex, regardless of their retinal or optic nerve condition.Several research groups around the globe are pursuing this goal [13–16,18]. In this context, the main goal of this work is to create and integrate the actual knowledge on neural function, psychophysics, signal processing and neural encoding modeling, and build a working pipeline which leads us towards further experiments and techniques that advance in the development of cortical visual prostheses. A general idea of the complete pipeline of a functional cortical prosthesis is composed of a video camera which receives the visual information, sends its to a signal processing device which sends orders to the neurostimulator that sends electrical pulse trains to the neural tissue accordingly to that commands (see Fig. 1) [13,14]. In the scenario of a NIH approved human clinical trials project named Development of a Cortical Visual Neuroprosthesis for the Blind [17], we now face the challenge of creating not only powerful but also flexible tools that overcome the limitations and needs of the current neurostimulation systems, allowing for new experimental trials.

Inspired by the success of cochlear implants, who greatly benefited from the developing and tuning of signal processing models according to psychophysics [19], we designed an end-to-end image processing and stimulation control

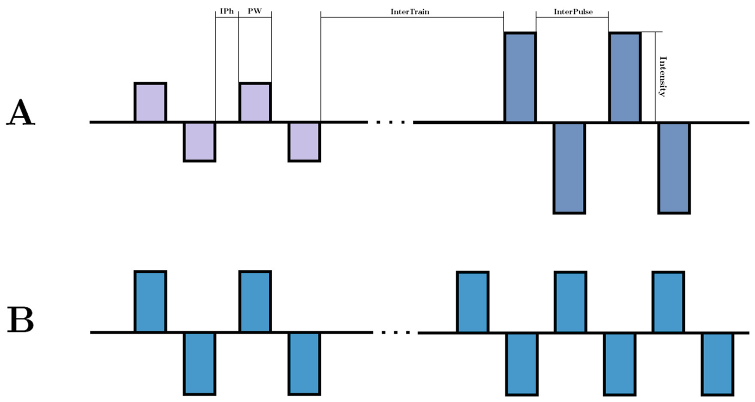

Fig. 2. Two main train modulation strategies are contemplated: Intensity modulation (A) and Frequency modulation (B).

workflow aimed to constitute a useful tool for neuroprosthesis research. In order to allow the designed system to incorporate bioinspired control capabilities of the electrical stimulation parameters (Amplitude of the phases, Pulse Width, Pulse Frequency, Inter Pulse, Inter Phase, Inter-Train, see Fig. 2), we created a neural encoding module which makes use of Deep Learning libraries Keras and Tensorflow [32,33], allowing us to create and make use of custom-defined or data-driven models of neural encoding of light patterns, simulating this way a retina-like visual processing, which output will be used for the stimulation control.

The possibility of reproducing neuronal activity present on the retina during natural vision has been studied with regards of electrical stimulation in epiretinal prosthesis [21], with promising results. In addition, techniques for computing population coding distances on retina have been proposed [22], along with visual perception simulation frameworks [20,23]. These results are encouraging, and the new methodologies could apply and be tested on the visual cortex.

In addition, diverse models of animal and human neural visual encoding systems of different nature have been created until today, targeting different processing stages, some of them focusing on the retina [24–27], which is the primary stage of visual processing.

2 System Overview

In this work, an end-to-end stimulation pipeline has been designed, integrating both hardware and software components into a flexible tool for visual neuroprosthesis research. This system, schematized in Fig. 3, is composed of several stages. First, a commercial USB camera device mounted on a pair of glasses captures the video signal that is received by a computer with a Linux operating system. We implemented the system with both a custom local computer and a Raspberry Pi model 3B+. The input images are then processed and sent to a model's

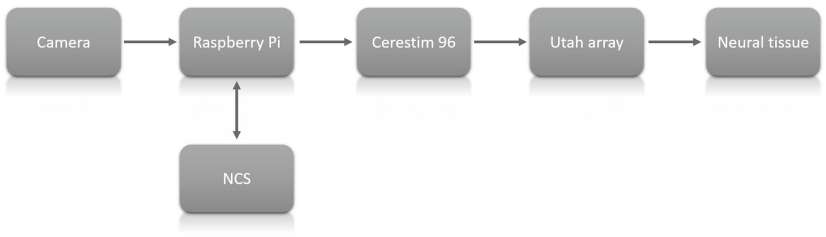

Fig. 3. Illustration of the main system's pipeline.

prediction module, an artificial retina, in our case. Finally, the model's output is interpreted as the main command for the neural stimulation (Cerestim96, Blackrock Microsystems, Inc., Salt Lake City, UT), which provides customized electrical pulse trains to the visual cortex through intracortical microelectrodes such as the Utah Electrode Array [31].

In order to handle the video stream, open source python libraries for scientific computing have been used: openCV scipy, numpy [28–30], along with state-of-the-art deep learning libraries: Tensorflow and Keras [32,33] as the tools to implement the neural coding previous to the stimulation control signals. In addition, we explored the possibility of deploying the visual coding models into a specialized deep learning acceleration hardware device [34], which relies on a Vision Processing Unit (VPU).

In the next section, we detail the function and features of the main blocks of the proposed system.

3 Software Interface Design

3.1 Modules Organization

In order to provide a modular, easy to use and extendable software corpus, we organized our python library, named Neurolight alpha, in the following structure:

Main_experiment. Contains the main thread on which the needed submodules are imported and the camera and stimulation devices configurations are set, before launching the "experiment" code.

Experiments. Each experiment is defined as a sequence of common steps: retrieving a new frame from the camera, preprocessing the incoming image (image normalization and resizing), optionally updating the video buffer (in case of a spatial-only retina model), performing the model's firing rate predictions, adapting the prediction to create stimulation commands, and sending those commands to the stimulator. The main workflow in the experiment module is detailed in Algorithm 1 (an example of an experimental workflow is described in Sect. 3.3).

Algorithm 1. Experiments module

```
Create train configurations
Select electrodes to use
Create encoding model
**while** processFlag **do**
  Obtain and process frame
  Update video buffer
  Model processing
  Stimulation command
  **if** $exitCondition == True$ **then**
    $processFlag \leftarrow True$
  **end if**
**end while**
```

VisionModels. This module allows to define Keras (TensorFlow) retinal processing models. It also contains model's prediction normalization functions, which are necessary to interface the neurostimulator correctly. In addition, it wraps the Neural Compute Stick API functions necessary to compile the designed Tensorflow models in order to deploy them into the Vision Processing Unit.

StimAPI. This module incorporates the main functions which builds the necessary messages that allows us to interact with the neurostimulator. Those are the basic commands blocks which are used by "StimControl".

StimControl. A higher level API that performs the communication operations needed to control the stimulator. Each StimControl function calls StimAPI one or more times in order to create and send functional commands.

NCSControl. This module can be used to accelerate the model's predictions upon the videostream on the Intel's Neural Compute Stick device. Contains functions for communication with the NCS device, loading models and performing inference over the input.

Utils. Contains various helper functions, as generating custom image filters(for example, Gaussian filters), or mapping the desired electrodes to the actual stimulator output channels.

3.2 Computational Neural Models and Image Preprocessing

The VisionModels module allows to define simple and complex, retina-like visual preprocessing models, which are defined as Keras Sequential models or Tensorflow Graphs. This models can be deployed using a CPU, GPU or specialized architectures, such as FPGAs. In this first design version, we prepared two modalities: spatial processing models and spatiotemporal processing models, where the defined filters are 2D and 3D, respectively.

Custom Linear-Nonlinear 2D/3D filter can be custom-defined or created using data-driven, machine learning techniques, as Linear-Nonlinear models or Convolutional Neural Networks (see Fig. 4).

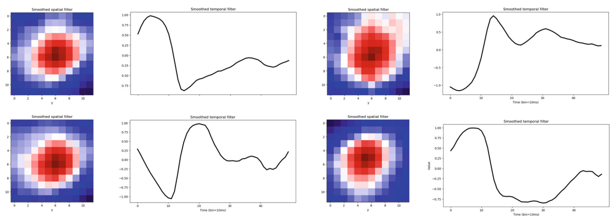

Fig. 4. Spatiotemporal filters can be defined and used as part of the pipeline. In the image, a selection of four of the linear part of the Linear-Nonlinear models after a rank-one decomposition performed with pyret [43]. The ganglion cell's retinal recordings were performed as mentioned in [24].

In order demonstrate the naturalistic stimulation control capability of our system, we fitted ganglion cell's firing responses to light patterns of different nature: full-field light flashes, checkerboard patterns, moving bars and natural scenes, following procedures similar of what is described in [24]. The natural scenes images were obtained from [35], and the rest of the stimuli was created by code scripts.

The ganglion cell's firing rates were fitted by means of a two-stage iterative process, in which each single neuron is modeled by means of a L2 regularized spatiotemporal Linear-Nonlinear process [36] (LN), which parameters are obtained with the Adam optimizer [37], which is a variant of the Gradient Descent optimization. The loss function utilized was a weighted sum of the mean squared error and the cross entropy between the biological retina's responses and the model's output.

In the first modeling stage, the input of the model consists on the flattened spatiotemporal visual stimulus that was projected into the retina during each time bin, and the output is the smoothed firing rate of the neuron as a response to the input [24]. The discrete time binning was 10 ms.

Due to the high dimensionality of the input (50 pixels $\times$ 50 pixels $\times$ 30 frames), the LN models created struggle to converge and are usually suboptimal. In order to tackle this, in the first stage we used a high regularization factor. This will promote that the model's parameters tends to zero in the spatial pixels which are out of the ganglion cell's receptive field, which is convenient in order to figure out which parts of the image are being encoded by the neuron. For the second modeling stage, we centered the model's target around the most relevant 15×15 pixels for each neuron, decreasing this way the number of parameters from 75000 to 6750, this is, 11x less parameters, leading the model to a more robust and faster convergence.

Once the Linear-Nonlinear models of the neurons are created, they can be loaded as the weights of a 1-layer Convolutional Neural Network either into a Keras sequential model (for quick deployment into the working pipeline) or TensorFlow graph, which will allow to compile it into a specific graph format to be used by a Neural Compute Stick (see Sect. 3.4).

Both the Keras and Tensorflow libraries allow for application-specific customization, by selecting the convolution stride or making spatial or spatiotemporal predictions over batches of images.

3.3 Neurostimulator Control and Stimulation Strategy

We developed and implemented a python version of the Blackrock Microsystems' API for the control of the CereStim96 neurostimulation device (briefly described in Sect. 3.1). This device allows for 16 simultaneous active channels, and 15 different pulse train configurations, that can be dynamically created (by overriding previous configurations on demand). After checking the correct operation of the device's current modules, a base pulse train configuration is defined. Then, modified versions of the base pulse trains are created, with different pulse intensity/frequency/pulse width values. This configurations, which shapes the pulse trains that will be delivered through the corresponding channels, are defined by the following parameters: Amplitude1, Amplitude2, PulseWidth1, PulseWidth2, InterPhase, InterPulse (see Fig. 2). Another key parameter to take in consideration its the InterTrain, this is, time between train pulses.

In this experiment, we select a list of electrodes which will be activated and map them into the actual channels which the device connects to.

After this, the camera configuration parameters are set, taking into consideration the dimensions of the input image, and the number of frames to buffer for the spatiotempoal processing. The retina model is defined by loading, reshaping and normalizing the ganglion cell's Linear-Nonlinear -or any other customized filter-based- model's weights into a Keras or Tensorflow model which will handle the convolution operations and strides. In the case of using a hardware acceleration device like the NCS for offloading the model's computations, we compile the desired model and load it into the device.

Once the main configurations are set, the main thread starts. Each input frame from the camera device is handled by openCV, resized and normalized, and stored into a buffer variable of the desired length. This buffer will be processed by the retina model, either in the main computer of in the acceleration device. The model's predictions are then normalized between 0 and 1 and matched to the closest of the 15 configurations selected for each electrode, which vary either in the intensity of frequency, depending on the desired train modulation strategy. Then, a single group stimulation sequence is generated for the corresponding electrodes and configurations and the command is sent to the stimulation device. If desired, a prompt windows will be updated, showing the camera input and stimulation information. After checking that the stimulation operation was properly performed, the next frame is obtained from the camera, and the whole process is repeated.

The stimulation strategies must be shaped by decisions of diverse nature: the actual knowledge of the psychophysics, computational modeling needs, software design decisions, hardware features and limitations. Two main strategies are contemplated currently: Amplitude Modulation and Frequency modulation (see Fig. 2), where the model's predictions for each electrode are mapped to the closest matching configurations, which are previously set and loaded into the stimulator device.Clinical studies will reveal the most optimal way of operating. [12,38–41].

3.4 Hardware Implementation

In order to explore the possibilities that dedicated hardware acceleration offers, we deployed the vision models into a Intel's Neural Compute Stick [34], a low-power consuming, edge-computing device designed to deploy Deep Learning models for inference, which incorporates an Intel® Movidius^{TM} Vision Processing Unit (VPU). This device is connected to the main computer, and the preprocessed image/video data its passed to it through an USB port, returning the model's predictions after the processing.

After creating single Linear-Nonlinear ganglion cell's models, they were loaded into a Tensorflow graph. The Linear-Nonlinear models, as described in Sect. 3.2 consists of a spatial or spatiotemporal filter with a nonlinear activation function which is convoluted through the image/video input, returning the predicted ganglion cell's firing rates, predictions which can be used for the stimulator control after proper normalization and configuration-matching (see Fig. 5). The created graph its prepared for inference-only mode and compiled into a compatible format to be used into the device.

4 Discussion and Future Work

Since many questions regarding the psychophysics of the phosphene generation are either still unanswered or in need of a more extensive exploration, cortical neurostimulators control tools must be smooth and easy to use and at the same time they have to permit to be adapted to the ongoing experimental findings as the theoretical and experimental hyopothesis are confirmed or discarded in the clinical research.

In this work, a cortical prosthesis control framework prototype is developed, having at its core the design principles of robustness and flexibility, allowing custom adaptation to the needs of clinical research. This functional working pipeline allows to incorporate both simple and complex computational neural encoding models of visual inputs (such as data-driven or custom linear-nonlinear models of retinal ganglion cells) for prosthesis control and to define different stimulation strategies, such as amplitude and frequency modulation, based on the implemented models. This framework has been designed and implemented into an experimental setup with a commercial neurostimulator that can be used on both animal an human research. The core pipeline modules, as the image capturing and preprocessing, the neural encoding module and the stimulator control API are based on open-source python libraries commonly used by the scientific community, which we believe is a fundamental feature for scientific tools and knowledge sharing.

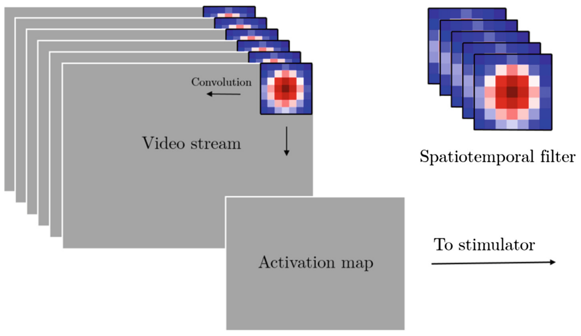

Fig. 5. Illustration of a spatiotemporal filter being applied to a video stream. The output of the model is an array of activation values that can be mapped to the stimulator's electrodes, after a proper normalization and matching with the pre-configured stimulus configuration values.

Among the weaknesses of the current system its the fact that its pipeline is sequential: every new image has to be processed before sending the image/video data for the model to return its output. After that, the stimulator has to finish its function before a new image its fetched. With this operation mode, the system is able to change the running stimulation parameters at 14 FPS. This bottleneck can be avoided by using threading, and its one of the main improvements to be made in future works. As a future planned improvement, Neurolight will allow to parallelize the image/video inputs pre-processing with the model encoding modules and the stimulation control, obtaining this way a better maximum system's performance in terms of FPS and more diverse possibilities for stimulation strategies development.

The fact that the electrical pulse trains are sent after the image processing-model prediction stage leads to a blinking stimulation strategy: train pulses are interleaved with an inter-train resting period. This way of stimulation on the visual cortex has been tested previously [39], and prevents the neural tissue to be permanently under the influence of external electrical fields, although the implications of this stimulation strategy has yet to be elucidated. In this matter, the inter-train interval necessary to generate a separated or continuous phosphene will be one of the main features of study in the clinical phases, along with the effects of temporal summation, and phosphene size and brightness.

Among the main challenges that a visual prosthesis designer faces is how to convey the most useful information trough the prosthesis. One of the most promising alternative pre-processing strategies is semantic segmentation [42] - which can already be implemented into our working pipeline by compiling a U-NET-like semantic segmentation CNN. The main idea of this approach is to simplify the transmitted visual information in a meaningful way, such that the complexity of the environment is diminished without disregarding the important information necessary for scene understanding and navigation.

In future works, we expect to implement several psychophysics modules which complement the tool and allow for a better fine-tuning of the whole system. Regarding the visual encoding models, it is hypothesized that retina-like image preprocessing could be beneficial for visual prosthesis [13], by performing a bioinspired feature extraction of visual information, although this remains unanswered. Along with the technical achievements made, new experiments need to be designed accordingly to provide answers. In this way, more complex, CNN-RNN based retina models which are proven to mimic the retinal encoding will be compiled and tested, and a tradeof between model's complexity and overall system's performance in terms of computing speed will be extensively studied.

We hope that the present work constitutes a step forward towards integrating knowledge from many scientific and engineering fields into a useful clinical research tool, and it is designed under the aim that neural engineers dream of: to help people achieving a level of neural function recovery sufficient to improve their life's quality.

Acknowledgments. This work is supported by the Programa de Ayudas a Grupos de Excelencia de la Región de Murcia, Fundación Séneca, Agencia de Ciencia y Tecnología de la Región de Murcia.

References

1. Wolpaw, J.R., et al.: Brain-computer interface technology: a review of the first international meeting (2000)
2. Davis, T.S., et al.: Restoring motor control and sensory feedback in people with upper extremity amputations using arrays of 96 microelectrodes implanted in the median and ulnar nerves. J. Neural Eng. **13**(3), 36001 (2016)
3. Nuyujukian, P., et al.: Cortical control of a tablet computer by people with paralysis. PLoS ONE **13**(11), e0204566 (2018)
4. Fattahi, P., Yang, G., Kim, G., Abidian, M.R.: A review of organic and inorganic biomaterials for neural interfaces. Adv. Mater. **26**(12), 1846–85 (2014)
5. House, W.F.: Cochlear implants. Ann. Otol. Rhinol. Laryngol. **85**(Suppl. 3), 3 (1976)
6. Weiland, J.D., Liu, W., Humayun, M.S.: Retinal prosthesis. Annu. Rev. Biomed. Eng. **7**(1), 361–401 (2005)
7. Mayberg, H.S., et al.: Deep brain stimulation for treatment-resistant depression. Neuron **45**(5), 651–660 (2005)
8. Sengupta, A., et al.: Red-shifted channelrhodopsin stimulation restores light responses in blind mice, macaque retina, and human retina. EMBO Mol. Med. **8**(11), 1248–1264 (2016)
9. Vassanelli, S., Mahmud, M.: Trends and challenges in neuroengineering: toward intelligent neuroprostheses through brain inspired systems; communication. Front. Neurosci. **10**, 438 (2016)
10. da Cruz, L., et al.: Five-year safety and performance results from the argus II retinal prosthesis system clinical trial. Ophthalmology **123**(10), 2248–2254 (2016)
11. Hornig, R., et al.: Pixium vision: first clinical results and innovative developments. In: Gabel, V. (ed.) Artificial Vision, pp. 99–113. Springer, Cham (2017). https://doi.org/10.1007/978-3-319-41876-6_8

12. Fernandez, E.: Development of visual neuroprostheses: trends and challenges. Bioelectron. Med. **4**(1), 12 (2018)
13. Normann, R.A., Greger, B.A., House, P., Romero, S.F., Pelayo, F., Fernandez, E.: Toward the development of a cortically based visual neuroprosthesis. J. Neural Eng. **6**(3), 35001 (2009)
14. Dobelle, W.H.: Artificial vision for the blind by connecting a television camera to the visual cortex. ASAIO J. **46**(1), 3–9 (2000)
15. Troyk, P., et al.: A model for intracortical visual prosthesis research. Artif. Organs **27**(11), 1005–1015 (2003)
16. Lowery, A.J.: Introducing the Monash vision group's cortical prosthesis. In: IEEE International Conference on Image Processing 2013, pp. 1536–1539 (2013)
17. Development of a Cortical Visual Neuroprosthesis for the Blind (CORTIVIS). ClinicalTrials.gov. Identifier: NCT02983370
18. Early Feasibility Study of the Orion Visual Cortical Prosthesis System. ClinicalTrials.gov. Identifier: NCT03344848
19. Shannon, R.V.: A model of threshold for pulsatile electrical stimulation of cochlear implants. Hear. Res. **40**(3), 197–204 (1989). https://doi.org/10.1016/0378-5955(89)90160-3
20. Golden, J.R., et al.: Simulation of visual perception and learning with a retinal prosthesis. J. Neural Eng. **16**, 025003 (2019)
21. Jepson, L.H., Hottowy, P., Weiner, G.A., Dabrowski, W., Litke, A.M., Chichilnisky, E.J.: High-fidelity reproduction of spatiotemporal visual signals for retinal prosthesis. Neuron **83**(1), 87–92 (2014)
22. Shah, N.P., Madugula, S., Chichilnisky, E.J., Shlens, J., Singer, Y.: Learning a neural response metric for retinal prosthesis (2018)
23. Beyeler, M., Boynton, G., Fine, I., Rokem, A.: pulse2percept: A Python-based simulation framework for bionic vision. In: Proceedings of the 16th Python in Science Conference, pp. 81–88 (2017)
24. Lozano, A., Soto-Sánchez, C., Garrigós, J., Martínez, J.J., Ferrández, J.M., Fernández, E.: A 3D convolutional neural network to model retinal ganglion cell's responses to light patterns in mice. Int. J. Neural Syst. **28**(10), 1850043 (2018)
25. Crespo-Cano, R., Martínez-Álvarez, A., Díaz-Tahoces, A., Cuenca-Asensi, S., Ferrández, J.M., Fernández, E.: On the automatic tuning of a retina model by using a multi-objective optimization. In: Artificial Computation in Biology and Medicine, Elche, Spain, pp. 108–118 (2015)
26. Mcintosh, L., Maheswaranathan, N., Nayebi, A., Ganguli, S., Baccus, S.: Deep learning models of the retinal response to natural scenes. In: Advances in Neural Information Processing Systems, Barcelona, Spain, vol. 29, pp. 1369–1377 (2016)
27. Yan, Q., et al.: Revealing fine structures of the retinal receptive field by deep learning networks (2018). (Lateral geniculate nucleus, V1, V4...). In our work, we focus on the first stage of visual processing: the retina
28. Bradski, G.: The openCV library. Dr. Dobb's J. Softw. Tools **25**, 120–125 (2000)
29. Jones, E., Oliphant, T.E., Peterson, P., et al.: SciPy: open source scientific tools for Python (2001)
30. Travis E, Oliphant. A Guide to NumPy. Trelgol Publishing, USA (2006)
31. Maynard, E.M., Nordhausen, C.T., Normann, R.A.: The utah intracortical electrode array: a recording structure for potential brain-computer interfaces. Electroencephalogr. Clin. Neurophysiol. **102**(3), 228–239 (1997). https://doi.org/10.1016/s0013-4694(96)95176-0

32. Abadi, M., et al.: TensorFlow: large-scale machine learning on heterogeneous distributed systems. In: Computer Science - Distributed, Parallel, and Cluster Computing, Computer Science - Learning (2016)
33. Chollet, F.: Keras (2015). https://github.com/fchollet/keras
34. Intel's Neural Compute Stick. https://movidius.github.io/ncsdk/ncs.html
35. Deng, J., et al.: ImageNet: a large-scale hierarchical image database. In: CVPR (2009)
36. Baccus, S.A., Meister, M.: Fast and slow contrast adaptation in retinal circuitry. Neuron **36**(5), 909–919 (2002)
37. Kingma, D.P., Ba, J.L.: ADAM: a method for stochastic optimization
38. Dobelle, W.H., Mladejovsky, M.G.: Phosphenes produced by electrical stimulation of human occipital cortex, and their application to the development of a prosthesis for the blind. J. Physiol. **243**(2), 553–576 (1974)
39. Schmidt, E.M., Bak, M.J., Hambrecht, F.T., Kufta, C.V., O'Rourke, D.K., Vallabhanath, P.: Feasibility of a visual prosthesis for the blind based on intracortical micro stimulation of the visual cortex. Brain **119**(2), 507–522 (1996)
40. Davis, T.S., et al.: Spatial and temporal characteristics of V1 microstimulation during chronic implantation of a microelectrode array in a behaving macaque. J. Neural Eng. **9**(6), 65003 (2012)
41. Foroushani, A.N., Pack, C.C., Sawan, M.: Cortical visual prostheses: from microstimulation to functional percept. J. Neural Eng. **15**(2), 21005 (2018)
42. Long, J., Shelhamer, E., Darrell, T.: Fully convolutional networks for semantic segmentation (2015)
43. Benjamin Naecker, N.M.: pyret: retinal data analysis in Python - pyret 0.6.0 documentation