An Embedded Real-Time Processing Platform for Optogenetic Neuroprosthetic Applications

Boyuan Yan and Sheila Nirenberg

***Abstract*— Optogenetics offers a powerful new approach for controlling neural circuits. It has numerous applications in both basic and clinical science. These applications require stimulating devices with small processors that can perform real-time neural signal processing, deliver high-intensity light with high spatial and temporal resolution, and do not consume a lot of power. In this paper, we demonstrate the implementation of neuronal models in a platform consisting of an embedded system module and a portable digital light processing projector. As a replacement for damaged neural circuitry, the embedded module processes neural signals and then directs the projector to optogenetically activate a downstream neural pathway. We present a design in the context of stimulating circuits in the visual system, but the approach is feasible for a broad range of biomedical applications.**

***Index Terms*— Embedded processor, neural signal processing, neural pathway modeling, optogenetic stimulation, digital light processing, neuroprosthesis.**

I. Introduction

OPTOGENETICS allows neurons to be selectively turned on or off with unprecedented precision [1]–[5]. This offers great opportunities for clinical applications because it provides a way to activate or inactivate specific neurons or classes of neurons in a malfunctioning or damaged circuit and re-engage them into normal activity. As such, the development of optogenetic tools offers new hope for patients suffering from neurological disorders [6] such as epilepsy [7]–[9] Parkinson's disease [10], [11], hearing impairments [12], or vision impairments [13]–[22].

The basic strategy of an optogenetic prosthesis is to bypass a damaged neural pathway and provide direct stimulation to downstream components in the pathway. To implement the strategy in a device, two components are needed: a processor, which processes neural signals in real time, to mimic the function of the pathway being bypassed, and a stimulator to optogenetically activate the downstream cells.

From the stimulator point of view, it is necessary to target optogenetic proteins with precise spatial and temporal resolution. Since these proteins require bright light [23], [24], standard high resolution devices such as LCD monitors are ineffective. While laser or LED-based illumination systems can meet the requirements of light intensity, they are not readily suitable for providing spatially-patterned stimuli. Digital light processing (DLP) projectors, however can achieve this: the key component, a chip called Digital Micromirror Device (DMD) [25], consists of an array of several hundred thousand micromirrors, which can change position at kHz frequencies. These offer the most flexibility in terms of the spatial and temporal modulation of the activating light.

In addition to an effective stimulator, the other essential component is an efficient processor to mimic the processing of the neural pathway being bypassed. To replace the function of the bypassed system, processors are required to simulate neural processing and signaling in real time. To achieve sufficient spatial and temporal resolution, it may involve the simulation of a large number of model cells at a temporal scale that is on the order of milliseconds, which is computationally intensive [26]–[28]. Besides the real-time computing constraint, the processor also needs to be small and consume less power so that the prosthetic device can be portable and energy efficient (so batteries last long enough for the device to be useful). These requirements make embedded processors with low power consumption, small size, and low cost a perfect candidate.

In this paper, we demonstrate the implementation of neuronal models in a platform consisting of an embedded system module and a portable DLP projector for high spatial and temporal resolution optogenetic stimulation. The platform is described in the context of driving visual circuits: briefly, the embedded module takes visual input (data from a camera), passes it through an array of model neurons to produce firing patterns that are similar to the patterns produced by the targeted downstream cells. The DLP projector then optogenetically actives the targeted downstream cells to fire in these same patterns. To allow processing in real time, models are simulated with a heterogeneous architecture composed of a general-purpose processor (GPP) and a digital signal processor (DSP). Although presented in the context of visual systems, the platform can be easily adapted for a broad range of biomedical applications.

The paper is organized as follows: In Section II, we describe our strategy for implementing a neuronal model in a digital signal processor. In Section III and Section IV, the

Manuscript received February 7, 2017; revised July 29, 2017 and September 25, 2017; accepted October 2, 2017. Date of publication October 16, 2017; date of current version January 8, 2018. This work was supported in part by the National Eye Institute under Grant R01 EY12978 and in part by the Department of Defense under Grant W81XWH-12-1-0231. *(Corresponding author: Sheila Nirenberg.)*

The authors are with the Department of Physiology and Biophysics, Weill Cornell Medicine, Cornell University, New York City, NY 10021 USA (e-mail: boy2004@med.cornell.edu; shn2010@med.cornell.edu).

Digital Object Identifier 10.1109/TNSRE.2017.2763130

hardware platform and software design are presented, respectively. In Section V, an illustrative example is provided to demonstrate the capability of the platform. Section VI gives a detailed discussion on the applicability and limitations.

II. Model

In this section, we describe our implementation of compact neuronal models in digital signal processors.

A. Firing-Rate Model

A variety of neuronal models have been proposed to mimic the processing performed by neurons in the nervous system. In this paper, we use the classic firing-rate model as an example (reviewed in [29]). This model [29] is described in a network setting, where both inputs and outputs are characterized by firing rates. Given a neuron receiving P synaptic inputs labeled by $i = 1, 2, \ldots, P$, the synaptic current entering the soma $I(t)$ in terms of the firing rates of the presynaptic neurons is determined by the following first-order differential equation

$$\tau \frac{dI(t)}{dt} = -I(t) + \sum_{p=1}^{P} w_p u_p(t - d_p), \tag{1}$$

where τ denotes the time constant, u_p denotes the firing rate of the pth input, and d_p and w_p denote the corresponding axonal conduction delay and synaptic weight, respectively. Given $I(t)$, the firing rate of postsynaptic neuron $\lambda(t)$ can be expressed as

$$\lambda(t) = \mathcal{F}(I(t) + I_0), \tag{2}$$

where $\mathcal{F}$ is called an activation function, and I_0 is a constant offset which sets the baseline firing rate of the model.

Firing rate models can be used to generate stochastic spike sequences from a deterministically computed rate. A spike train is typically described by an inhomogeneous Poisson process, which involves a time-dependent firing rate $\lambda(t)$ [29]. If n is the observed number of spikes during a given time interval Δ, the probability of such an event is given by a Poisson distribution

$$P(n|\lambda(t)) = \frac{(\lambda(t)\Delta)^n}{n!} exp(-\lambda(t)\Delta). \tag{3}$$

The Poisson process provides a useful approximation of stochastic neuronal firing, and we show how we use it to generate Poisson spikes trains below.

More generally, the rate model given by (1)(2)(3) can be used as a simplified functional model to describe the stimulus-response characteristics of spiking neurons in early sensory pathways. In this case, the input $u_p(p = 1, \ldots, P)$ represents the pth component of an external sensory stimulus. For example, in the visual system, the inputs could represent a number of pixels that fall on the receptive field of a neuron.

B. Implementation of Firing-Rate Models in the Discrete-Time Domain for Real-Time Applications

The firing-rate model above is described by a first-order delayed differential equation, which is a continuous time model. For many practical applications, the inputs are typically represented by discrete-time stimulus sequences (e.g, a sequence of frames from camera), and the storage of any time function in a computer or digital signal processor can only be as a discrete function. Therefore, a discrete-time system model is often needed from the implementation point of view.

1) Difference Equation: Discrete-time systems are described by difference equations. In general, a Qth order discrete-time system with a number of P inputs is described by the input/output difference equation

$$I[k] = -\sum_{q=1}^{Q} a_q I[k-q] + \sum_{p=1}^{P} \sum_{m=0}^{M} b_{pm} u_p[k-m], \tag{4}$$

where k is the integer-valued discrete-time index, u_p is the pth input, I is the output, and a_q and b_{pm} are constant coefficients. The equation shows a Qth order recursion process, i.e., the next value of the output is computed from the Q previous values of the output and the $P(M+1)$ values of the input at the cost of $O(Q + P(M+1))$. There is a trade-off between accuracy and simplicity. Models with a larger Q and M typically provide a more accurate description of the responses characteristics but they are also more expensive to calculate.

The time interval between any pair of consecutive sample times is the signal's sampling period, T_s, and the sampling rate F_s is the reciprocal of the sampling period

$$F_s = 1/T_s. \tag{5}$$

2) Spike Generation: For an inhomogeneous Poisson process, the probability of observing exactly n spikes in a particular interval $\triangle$ is given by (3).

For sufficiently small intervals $\triangle$, the average number of spikes can be approximated by $\langle n \rangle = \lambda\triangle$. In addition, the interval can be reduced until the probability of occurrence of more than one spike in the interval is sufficiently small that it can be ignored. In this case, the probability of a spike occurring in a brief time interval is equal to the product of the instantaneous firing rate during that interval and the length of the interval

$$P(n = 1|\lambda) = \lambda\triangle. \tag{6}$$

Therefore, in order to generate spikes, we need to divide the sampling periods (e.g., frame periods), which are typically large (e.g., T_s = 16.7ms, 33.3ms, 66.7ms for F_s = 60Hz, 30Hz, 15Hz, respectively) compared to the duration of spikes (typically 1-2ms). Given the sampling period of the input stimuli T_s, we choose J large, and divide each period into J bins (each of width $\triangle = T_s/J$).

Take the period of the kth frame (t_{k-1}, t_k) as an example. Given $I[k-1]$ at t_{k-1} and $I[k]$ at t_k, we perform a linear interpolation to calculate $I[k-1+j/J]$ at $t_{k-1+j/J}$, where $j = 1, \cdots, J-1$. For $j = 1, \cdots, J$, we calculate the firing rate at each bin as

$$\lambda_{k,j} = \mathcal{F}(I[k-1+j/J] + I_0), \tag{7}$$

where $\mathcal{F}()$ is the static nonlinear function, and I_0 is a constant offset which sets the baseline firing rate of the model.

Algorithm 1 Implementation of Firing-Rate Model to Recursively Process a Discrete-Time Input Sequence $u_p[k]$

Input: $u_p[k](p = 1, \ldots, P)$
Output: $n_{k,j}(j = 1, \ldots, J)$

1 $I[k] = -\sum_{q=1}^{Q} a_q I[k-q] + \sum_{p=1}^{P} \sum_{m=0}^{M} b_{pm} u_p[k-m]$
2. Add a constant offset $I[k] = I[k] + I_0$
3. Given $I[k-1]$ at t_{k-1} and $I[k]$ at t_k, perform a linear interpolation to calcuate $I[k-1+j/J]$ at $t_{k-1+j/J}$, where $j = 1, \cdots, J-1$.
4. For bins j $(j = 1, \ldots, J)$
   Calculate the firing rate $\lambda_{k,j} = \mathcal{F}(I[k-1+j/J] + I_0)$
   Draw a Bernoulli random variable $\mathcal{X}$ with probability $\lambda_{k,j}\triangle$
   Set $n_{k,j} = 1$ if $\mathcal{X} = 1$, or set $n_{k,j} = 0$ if $\mathcal{X} = 0$

A bit array $n_{k,j}$ $(j = 1, \cdots, J)$ is used to store the spikes corresponding to the kth frame: for the jth bin $(j = 1, \cdots, J)$, we draw a Bernoulli random variable $\mathcal{X}$ with probability $\lambda_{k,j}\triangle$. If there is a spike $(\mathcal{X} = 1)$, we set $n_{k,j} = 1$; otherwise $(\mathcal{X} = 0)$, we set $n_{k,j} = 0$.

For each bin, the cost mainly involves a linear interpolation, a static nonlinear function evaluation, and a random number generation. The linear interpolation takes two multiplications and one summation, the static nonlinear function can be implemented as a lookup table, and the random number generator can be implemented by a pre-stored random number sequence. For each frame, the cost of the spike generation step is dominated by $O(cJ)$, where J is the number of bins per frame, and c is the number of operations involved for each bin, typically $c \ll J$.

3) Algorithm: The implementation of firing-rate model to recursively process a discrete-time input sequence $u_p[k]$ is shown in Algorithm 1. Since the cost for the first two steps is $O(Q + P(M+1))$, and the cost for the last two steps is $O(cJ)$, it takes a total cost of $O(Q + P(M+1) + cJ)$ to process one frame in general.

III. Hardware Platform

As shown in Fig. 1, the hardware platform (less than $1000) consists of a camera (sensor), an embedded system module (processor), and a DLP projector (stimulator).

In the context of visual systems, images taken by the camera are converted into sequences of binary patterns by the embedded system module. The DLP projector then converts the patterns into spatially structured light pulses, which are delivered to drive light-sensitive proteins in the targeting cells.

A. Camera

In general, as the input of the platform, a sensor detects various environmental stimuli, and the detected signal is typically sampled at a constant rate. In the context of visual systems, we use a Caspa VL camera board (Gumstix Inc., San Jose, CA) as the sensor as shown in Fig. 1.

The camera has a 3.6mm fixed focal length lens with IR cut filter so that it receives only visible spectrum light, and a CMOS imager (Aptina MT9V032, 752(H)×480(V), 60 FPS).

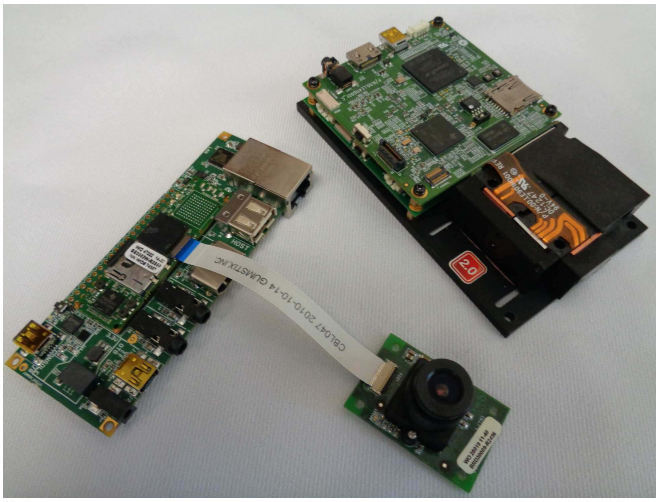

Fig. 1. The hardware platform consists of an embedded system module with a camera (left), and a portable DLP projector (right).

The active imager size is 4.51mm (H)×2.88mm (V), and the size of each pixel is $6.0\mu m \times 6.0\ \mu m$. Therefore, the angle of view is 64.1°(H)×43.6°(V), and each pixel corresponds to approximately a visual field of 0.1°.

B. Embedded System Module

A tiny low-power computer on module (COM) is used in the platform. The module (Overo Water, Gumstix Inc., San Jose, CA), roughly the size of a stick of gum, includes an embedded processor (OMAP3530, Texas Instruments, Dallas, TX), an onboard memory package (Micron 512MB DDR low-power DRAM and 512MB NAND flash memory), a highly integrated power-management integrated circuit, and an onboard card-slot for system storage expansion using microSD cards.

Owing to its small size, the module itself does not have any Standard I/O connectors. There is a Hirose 27-pin camera connector that connects a Caspa camera to the board, and two small 70-pin AVX connectors that mate with an expansion board that provides the usual connections (Ethernet, USB, HDMI video, etc). As shown in Fig. 1 (left), the Overo COM is mounted on a Tobi expansion board. Note that customized expansion boards could be built to achieve much smaller size by keeping only components necessary to the project.

The OMPA3530 processor is designed to provide high performance real-time video, image, and graphics processing while maintaining low power consumption. The processor integrates a 720MHz ARM Cortex-A8 general purpose processor (GPP), and a 520MHz digital signal processor (DSP) (TMS320C64x, Texas Instruments). For most applications, the GPP takes care of system management, command, and control. The DSP, on the other hand, is optimized for intensive signal processing in real time. The dual processor architecture thus provides a means to coordinate both processors to seamlessly leverage the unique capabilities of each.

C. DLP Projector

In recent years, as optogenetic technology rapidly develops, optical stimulation has become an alternative to

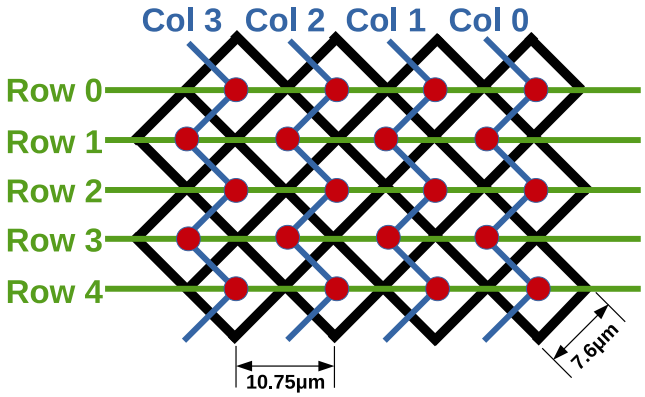

Fig. 2. The organization of the diamond pixels (square rotated 45 degrees). The row and column indices for each pixel are based on the addressing of the corresponding memory bit (represented by red dots). Since the distance between two neighboring columns is twice the distance between two neighboring rows, the 608 × 684 pixel array has an aspect ratio of 608:342.

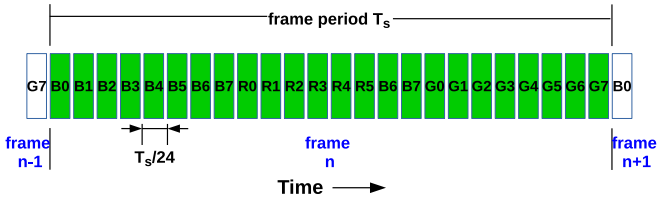

Fig. 3. The one to one mapping of RGB input bits to the sequence of output light pulses per pixel per frame. For example, the B5 pulse is associated with the bit 5 of byte Blue. Note that the duration of each light pulse is equal to the frame period T_S divided by 24.

electrical stimulation. Here, we present an optogenetic stimulator based on a portable DLP projector to achieve not only high light intensity but also high spatial and temporal resolution. The projector (DLP LightCrafter, Texas Instruments) shown in Fig. 1 (right) is a compact evaluation module for integrating projected light into industrial, medical, and scientific applications. The module has three main components: a digital micromirror device (DMD) and its control circuitry, a light module, and an output lens module.

1) Digital Micromirror Device: The DMD contained in the module is a 0.3-inch WVGA chip (DLP3000, Texas Instruments). It consists of a 608×684 array of micromirrors with diamond pattern geometry, each 7.6μm×7.6μm. Fig. 2 shows the organization of the diamond pixels. The module also includes control circuitry to receive input images over an HDMI connection and presents them on the DMD.

Importantly, the DMD supports a high speed monochrome pattern mode, in which each micromirror can be individually switched on or off at up to 1440Hz. In this mode, only one of the three colors (red, green, or blue) can be used. As shown in Fig. 3, the RGB data format for each pixel is interpreted as follows: bits 0-7 of byte Blue correspond to patterns 0-7, bits 0-7 of byte Red correspond to patterns 8-15, and bits 0-7 of byte Green correspond to patterns 16-23. The display sequence of the patterns is 0-23.

There are a number of 24 binary patterns per frame period: given a frame rate F_s, the pattern rate (denoted by F_p) is

$$F_p = 24 \times F_s, \tag{8}$$

TABLE I
ALLOWABLE HIGH SPEED BINARY PATTERN COMBINATIONS

Frame rate F_s (Hz)	Frame period T_s (ms)	Pattern rate F_p (Hz)	Pattern period t_p (ms)
60	16.7	1440	0.7
30	333	720	1.4
15	66.7	360	2.8

and the pattern period (denoted by t_p) is equal to the reciprocal of the pattern rate

$$t_p = 1/F_p. \tag{9}$$

Table I gives several allowable high speed binary pattern combinations.

2) Light Module: There are 3 LEDs in the light engine (peak wavelengths: 460nm (blue), 515nm (green), and 617nm (red)). Each LED has an optical collimator to collect the wide beam of light from the LED to produce a narrower beam. A set of dichroic mirrors recombine the different colors of light into one collinear beam. The recombined light passes through a fly-eye and condenser lens to provide uniform illumination to the DMD. When the micromirrors are in the on-position, the light is reflected through the projection lens; otherwise, the light is directed towards a light absorbing barrier. The power loss of DMD is characterized by the efficiency in visible light (420 to 700 nm): window transmission 97%, micromirror reflectivity 80%, array diffraction efficiency 86%, and array fill factor 92%.

The output intensity of each LED can be precisely controlled by adjusting the current through the LED. When at room temperature, the maximum current allowed is dependent on the cooling system: 633mA for passively cooled systems (no extra heat sinks or fans) and 1.5A for actively cooled systems (extra heat sink and fan).

The projection lens in the light engine is intended for projecting images onto large areas: the maximum focus distance is 2169mm and the image diagonal size is 60inch. To produce a smaller image that would provide the spatial resolution and intensity needed for optogenetic stimulation, additional converging lenses need to be placed in front of the projection lens [30]. For example, with an additional lens (focal length: 15mm, aperture: 3mm) placed a distance of 10mm, the image size of each DMD pixel is 11μm and the light intensity at the maximum current allowed for passively cooled systems (633mA) could reach 1.4mW/mm^2(460nm), 1.1mW/mm^2(515nm), and 1.2mW/mm^2(617nm), respectively. The light intensities delivered are sufficient for many channelrhodopsin variants to produce precise temporal resolution [23], [24]. For example, a new channelrhodopsin with fast kinetics and high light sensitivity, Chronos, reliably drove 100% spiking at light powers as low as 0.05mW/mm^2(470nm).

For passively cooled systems, the DLP evaluation module consumes power up to 12.5W (5V, 2.5A). The power is mostly consumed by the three LEDs (∼3.5W each at the maximum current of ∼0.7A). For optogenetic applications, the consumption of the DLP module can be reduced to ∼5.5W as only one LED is turned on in the monochrome mode.

Together with the embedded system module (3.3V, 0.4A), the whole platform consumes ~7W, which is less than the power typically consumed by a laptop computer (20-60W).

IV. Software Design

In this section, we describe the software aspects of the platform including embedded operating system, preprocessing for model cell placement and geometrical mapping, and real-time processing by heterogeneous computing.

A. Embedded Operating System and Cross-Compilation

In the OMAP3530 processor, since the GPP takes care of the control and the DSP is specialized for real-time computation, a complete operating system is typically installed on the GPP, and a simple Basic Input/Output System (BIOS) is sufficient for the DSP.

In the platform, we use the Ångström distribution (a Linux distribution for embedded devices) as the GPP operating system and TI DSP/BIOS real-time operating system on the DSP. The embedded Linux distribution and executable codes for the application are created on a host machine by cross-compilation. A cross-compilation produces executable codes for a target platform different from the host platform (where the compiler is running). This is necessary for building software packages for embedded platforms where compilation is infeasible due to limited resources.

B. Model Cell Placement and Geometrical Mapping

In the context of visual systems, the processor functions as an input/output model of the upstream visual pathway: images taken by the camera are converted into spatio-temporal patterns of light, delivered by the DLP to drive light-sensitive proteins in the downstream targeting cells. As a result, there are two streams of images: input images and output patterns, and it is necessary to establish a spatial correspondence between an input image and its output counterpart.

1) Spatial Range and Resolution of Optogenetic Stimulation: The spatial range is typically determined by the extent of neural tissue to be targeted. For visual systems, the targeting neural tissue could be retina, or other downstream structures that are responsive to visual input. Typically, cells in each structure in the pathway can be seen as contributing to a map of the visual field (retinotopic map) [31], and there is a correspondence between a cell location in the structure and a location in the visual field. For convenience, we use the visual angle as a way to reference the size of optogenetic stimulation on the targeting neural tissue.

For illustration purposes, we assume a circular targeting area, and the visual angle of its diameter is denoted by $\triangle_{out}$. The maximum resolution of the optogenetic stimulation is determined by the image size of a single output pixel of the DMD, and the visual angle of its side length is denoted by $\triangle_{out,1}$. A lower spatial resolution can be achieved by using a $s \times s$ ($s > 1$) pixel group as a super-pixel, and the visual angle of the side length of a super-pixel is denoted by $\triangle_{out,s} = s\triangle_{out,1}$.

The size of super-pixel affects the activation of neurons. For example, if the size is too small (i.e., only a small surface area of cell membrane is illuminated), the activated ion channels may not be sufficient to trigger an action potential. On the other hand, if the size is too large (i.e., multiple cells are covered), it causes a decrease in spatial resolution. Assume the density of targeting neurons is d per unit length (linear density), to achieve stimulation at cellular resolution, the length of a super-pixel should be close to the reciprocal of the linear density of neurons in the 2-dimensional surface ($1/d$). Nevertheless, for sensory systems, neighboring neurons often have similar receptive fields and their responses to natural stimuli are correlated. This makes it possible to use a larger super-pixel to drive neighboring neurons as a group without sacrificing a lot of spatial resolution.

2) Spatial Coordinate System of Input and Output Images: Often, the most convenient method for expressing locations in an image is to use pixel indices. The image is treated as a grid of discrete elements, ordered from top to bottom and left to right. For pixel indices (i,j), the row i increases downward, while the column j increases to the right. Pixel indices are integer values, and range from 1 to the length of the row or column. This is exactly how pixels are identified in each image.

However, while both input and output images are arrays of pixels, they may have different spatial scales or orientations. For example, in terms of spatial scales, the pixels of the input image (from the camera) and output image (from the DLP projector) may correspond to different sizes of visual field. In terms of orientation, the camera uses an orthogonal pixel array configuration; the DLP projector uses a diamond pixel array configuration.

Therefore, in order to establish the correspondence between input and output pixels, we adopt another method for expressing locations in an image, which uses a system of continuously varying coordinates (x,y) rather than discrete indices (i,j). In a spatial coordinate system like this, locations in an image are positions on a plane, and they are described in terms of x and y (not row and column as in the pixel indexing system). This means x and y can be any real numbers instead of integers. Thus, the correspondence between input and output pixels can be determined by describing both input and output images with a spatial coordinate system, in which each point is given a unique coordinate (x,y) in degrees of visual angle. Assume the coordinates of the image centers are (0,0), the coordinates of all the pixels can be calculated in degrees of visual angle.

3) Model Cell Placement: Given the size of a super-pixel $\triangle_{out,s}$ as the spatial resolution, the output image can be divided into an array of non-overlapping adjacent super-pixels. The number of super-pixels falling into the range determines the total number of processing units

$$N = (\pi \triangle_{out}^2/4)/\triangle_{out,s}^2, \tag{10}$$

and the coordinates of the super-pixel centers determine the locations of the processing units (x_i,y_i), where $i = 1, 2, \ldots, N$. In the context of visual systems, the processing units are model cells and the correspondence between input and output pixels can be determined by the receptive field of

the cell (i.e., the particular region of the visual field in which a stimulus will trigger the firing of that cell).

The receptive field is typically specified by a $r \times r$ array of squares centered at the location of the model cell, and each square spans a visual angle $\triangle_{rf,1}$. The coordinates of the square centers can be calculated from the coordinates of the model cell. For example, given a model cell located at (x_i, y_i), the coordinates of the centers of r^2 squares are given by

$$(x_i + k_x \triangle_{rf,1}, y_i + k_y \triangle_{rf,1}), \tag{11}$$

where $k_x = -(r-1)/2 + c$, $c = 0, \cdots, r-1$, and $k_y = -(r-1)/2 + d$, $d = 0, \cdots, r-1$. Given the coordinates of the centers, we can determine which input pixels encompass these centers, and take the values of the pixels as the corresponding stimuli to the model cell. If the square size is larger than the pixel size $\triangle_{rf,1} > \triangle_{in,1}$, the above process essentially corresponds to a downsampling of the input image.

4) Preprocessing to Generate Lookup Table: The correspondence between input and output pixels for each processing unit is established as follows: given a model cell, the input is specified by a set of pixel indices of the camera, and the output is specified by a set of pixel indices of the DMD.

In the implementation, a preprocessing process is used to calculate the mapping of input and output pixels onto each model cell, and the results are stored in a lookup table. For each model cell, there is an entry in the lookup table, which contains a set of input pixel indices and a set of output pixel indices. In the real-time processing, the lookup table can be consulted to quickly calculate the responses of each model cell based on the values of corresponding input pixels and then map the responses to the corresponding output pixels.

C. Real-Time Processing by Heterogeneous Computing

As mentioned in Section III, the processor integrates a general purpose processor (GPP) and a digital signal processor (DSP). In this section, we show how to take advantage of the heterogeneous system architecture for real-time processing.

Briefly, the DSP is reserved for intensive neural processing as an input/output model of the upstream visual pathway. The GPP takes care of the control and coordination, which sends inputs from the camera to the DSP and sends outputs from the DSP to the projector. The flow is shown in Fig. 4. There are two loops: one runs on the GPP and the other runs on the DSP. The DSP loop is essentially nested inside the GPP loop. The loops repeat forever until either the user issues a stop command or a preset number of frames are processed. To meet the real-time constraint, the execution time of the GPP loop for one iteration should be less than one frame period T_s.

At each iteration, the GPP grabs a new frame from the camera and preprocesses it. The preprocessing typically involves cropping away parts of the image that would fall out of the range of optogenetic stimulation and rescaling the light level and contrast range to fall within the operating range of the model cells.

After preprocessing is finished, the GPP waits for the DSP to finish the processing of the previous frame. Upon notification of completion, the GPP transfers the results of the previous frame from shared memory to local memory. Immediately, the GPP writes the preprocessed new frame to the shared memory, and then notifies the DSP that a frame is ready to be processed. By sending a new frame to the DSP right after the previous frame is processed, we can minimize the DSP's idle time and thus take maximum use of it.

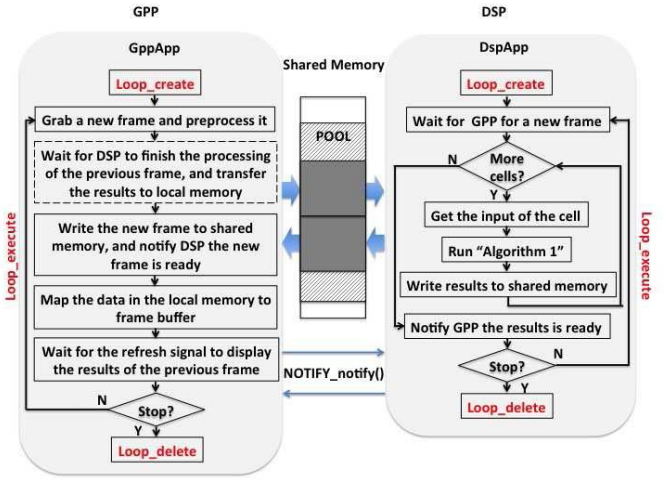

Fig. 4. The flow of real-time processing based on the GPP-DSP heterogeneous computing system. Note that the step in the dashed box on the GPP side is skipped at the first iteration as there is no frame under DSP processing at the beginning. Since the local memory locations to store the results are zero-initialized, nothing will be displayed at the end of the first iteration.

Upon notification of a new frame, the DSP loops through all the model cells: for each model cell, the input pixel indices are determined by consulting the lookup table. Given the values of the input pixels, the responses of each model cell can be calculated by running Algorithm 1. Given the frame period T_s, we choose $J = 24$, and divide the period into 24 bins. We use a 24-bit binary number to store the spike train of the model cell in response to the current frame, where a 1 at the ith bit indicates a spike in the ith bin ($i = 0, \cdots, 23$). The calculated spike trains are stored in the shared memory. Once all the results are available, the DSP sends a notification to the GPP.

On the side of GPP, after a new frame is sent to the DSP, it starts to prepare to display the results of the previous frame by mapping the data in the local memory to frame buffer. Specifically, the GPP loops through all the model cells to write the appropriate bytes into the video memory to define the image. For each model cell, the output pixels can be determined by consulting the lookup table, and the spike train of the model cell represented by a 24-bit binary number is written into RGB data format as described in Section III-C. Briefly, bits 0-7 are written into bits 0-7 of byte Blue, bits 8-15 are written into bits 0-7 of byte Red, and bits 16-23 are written into bits 0-7 of byte Green.

Once the frame of data is ready, the GPP sends it to the DLP upon the arrival of the next refresh signal, and then returns to the start of the loop to process the next frame. This imposes the real-time constraint: each frame needs to be processed within the time limit of one frame period T_s.

In the high speed pattern mode, the DLP decomposes the 24-bit RGB frame into 24 binary patterns and display the patterns in sequence. Note that a double buffering is employed in the GPP implementation: one part of memory in the frame buffer is used to display the current frame, and the other part of memory is used to store data for the next frame. The double buffering allows for a new frame to be written without disturbing the frame currently being displayed.

V. Results

In this section, we demonstrate the design and performance of the embedded real-time processing platform in the context of visual systems.

A. Model Cell Placement and Geometrical Mapping

As described in Section IV, a preprocessing process is used to determine model cell placement and geometrical mapping. In the illustrative example, we assume the spatial range is a circular retinal area corresponding to a visual angle of 10° ($\triangle_{out} = 10°$), and assume a maximum spatial resolution of 0.04°.

The maximum spatial resolution is determined by the visual angle of a single output pixel $\triangle_{out,1}$. A lower spatial resolution can be achieved by using a $s \times s$ ($s > 1$) pixel group as a super-pixel. Assuming a super-pixel formed by a 4×4 pixel group is being used, the visual angle of a super-pixel is 0.16° ($\triangle_{out,4} = 4\triangle_{out,1}$). Given the spatial range of 10° and the spatial resolution of 0.16°, the total number of model cells is around $N = 3045$, which is determined by the number of super-pixels falling into the spatial range according to (10).

Next, we need to map input and output pixels onto each model cell. The geometrical mapping for a model cell is illustrated in Fig. 5. The background pattern depicts the orthogonal pixel array of the camera overlaid on the diamond pixel array of the DLP projector. Each little square represents a pixel of the camera corresponding to a visual field of 0.1° ($\triangle_{in,1} = 0.1°$). Each little rhombus (square tilted at a 45 degrees angle) represents a pixel of the DLP projector corresponding to a visual field of 0.04° ($\triangle_{out,1} = 0.04°$).

A super-pixel formed by a 4×4 pixel group is shown at the center of the figure (represented by a larger rhombus). The super-pixel is about 1.6 times the size of the camera pixel. The corresponding model cell is placed at the center of the super-pixel. The receptive field of the model cell is specified by a 3×3 array of squares (represented by 9 larger squares), and we assume each square spans a visual angle of 0.27°($\triangle_{rf,1} = 0.27°$). The centers of all the squares (represented by red dots), calculated from (11), are sampling locations. The values of the camera pixels sampled at these locations are taken as the the stimuli to the model cell. In this example, since the square size $\triangle_{rf,1}$ is larger than the pixel size $\triangle_{in,1}$, the above process essentially corresponds to a downsampling of the input image.

To verify the geometrical mapping, we present a drifting vertical grating at 30 Hz frame rate for a duration of 20s, and examine the responses of model cells. The grating has a temporal frequency of 3Hz and a spatial frequency of 0.5cpd (the spatial period of the grating is 2°, which has a width of 20 input pixels as $\triangle_{in,1} = 0.1°$). The spatial temporal responses of model cells during one frame period of 33ms is shown in Fig. 6.

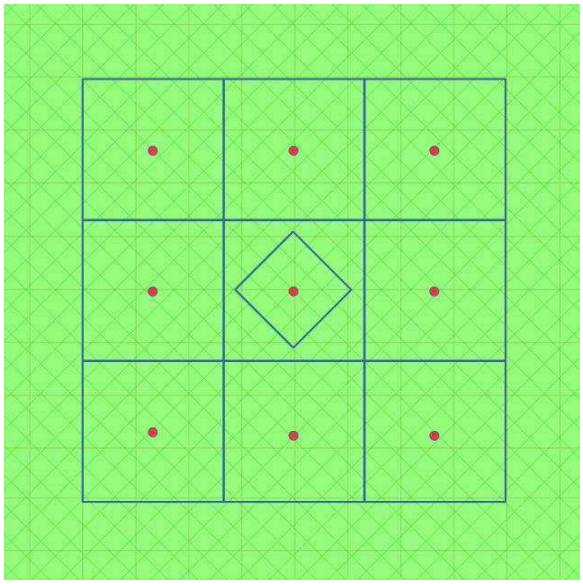

Fig. 5. An illustration of model cell placement and geometrical mapping: each little square represents a pixel of the camera (0.1°); each little rhombus represents a pixel of the DLP projector (0.04°); the larger rhombus at the center represents a super-pixel of the DLP projector ($textsf0.textsf16°$); the 3×3 array of larger squares (0.27°) specify the receptive field of the model cell located at the center of the super-pixel.

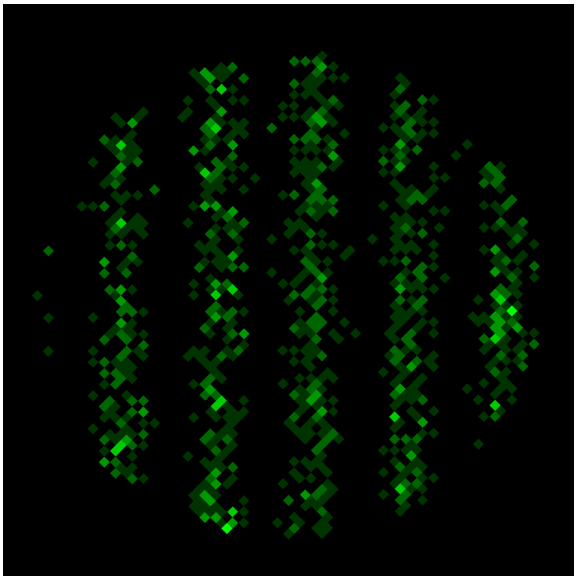

Fig. 6. The spatial temporal responses of models cells during one frame period of 33ms. The circular area corresponds to a visual angle of 10°, and each rhombus pixel represents a 4×4 super-pixel. The number of spikes occurring during the frame period is represented by 6 different intensity levels of the pixels: the complete darkness represents 0 and the brightest represents 5.

Given the geometric mapping, there should be certain phase relationships between the responses of model cells to the drifting vertical grating stimulus. For example, two model cells separated horizontally by one (half) of a spatial period should

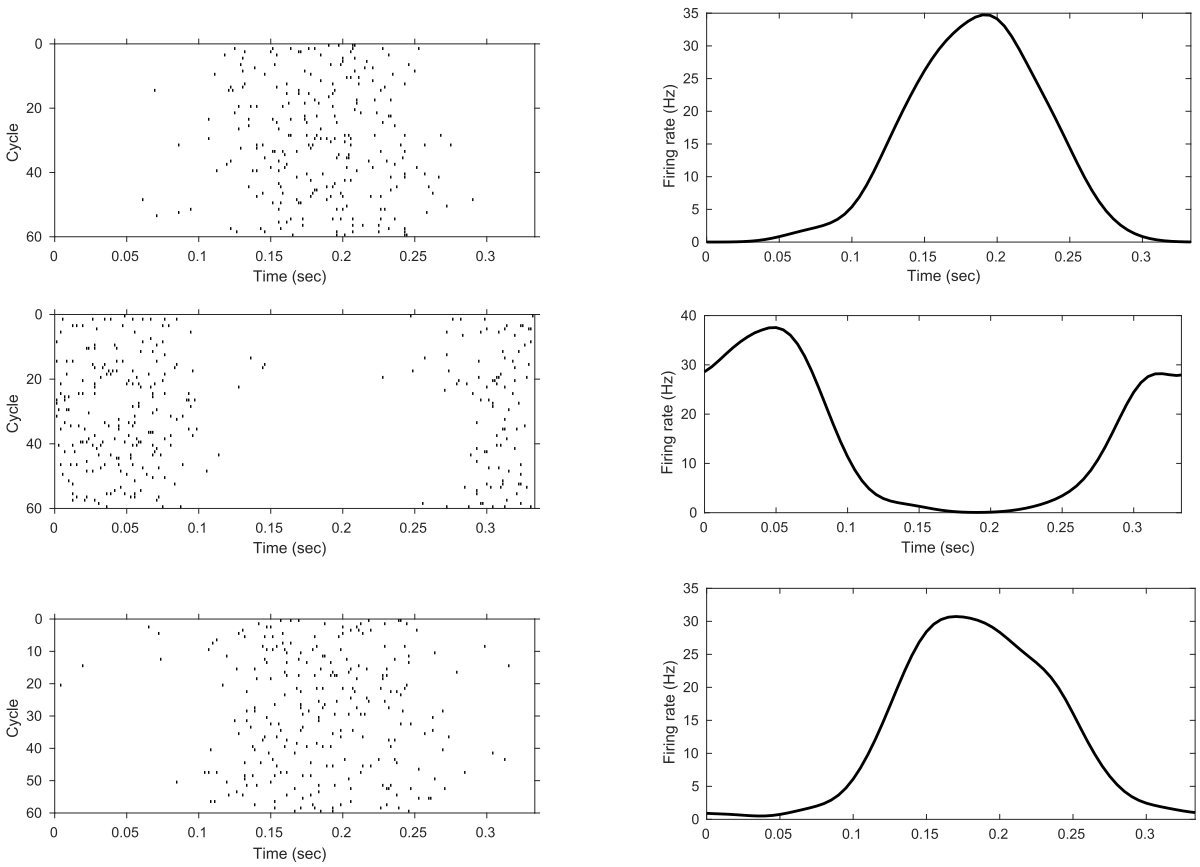

Fig. 7. The distribution of spike times of the three neighboring sampled cells (upper, middle, lower) over the temporal period of 0.33s. In each row, the left plot shows firing times of the model cell wrapped to one cycle of the stimulus, and the right plot shows peristimulus time histograms of the model cell, which represent the firing rate averaged across the 60 cycles.

be in-phase (anti-phase). In the current example, a super-pixel is formed by a 4×4 pixel group ($\triangle_{out,4} = 0.16°$), and the diagonal of the super-pixel is about 0.23°. Since the spatial period of the grating is 2°, the period is approximately as wide as 9 super-pixels horizontally in the diamond pixel array of the DLP projector. To verify the phase relationship, we uniformly sample 1 out of 4 super-pixels (cells) in a row of the diamond pixel array so that the center-to-center distance between two neighboring sampled super-pixels is about 0.9°, corresponding to a phase shift of 0.9π.

The responses of three neighboring sampled model cells are given in Fig. 7. Since the temporal frequency of the periodic stimulus is 3Hz, the 20s duration includes 60 cycles. Fig. 7 presents the distribution of spike times of the three cells (upper, middle, lower) over one temporal period of the stimulus (333ms). In each row, the left plot shows firing times of the model cell wrapped to one cycle of the stimulus, and the right plot shows peristimulus time histograms of the model cell, which represent the firing rate averaged across the 60 cycles (smoothed by a Gaussian function with 17ms SD). As expected, the phases of spikes of the model cells are locked to the 3Hz periodic stimulus, the second cell is approximately antiphase with respect to the first cell (phase difference: $\sim 0.9\pi$), and the third cell is approximately in phase with respect to the first cell (phase difference: $\sim 1.8\pi$).

B. Real-Time Performance and Scalability

Last, we test the scalability of the processing platform under real-time constraint. The scalability of the processing platform is measured by the number of model cells that can be processed in real-time. A larger number of model cells enable a larger stimulation area given a fixed spatial resolution; alternatively, a larger number of model cells enable a higher spatial resolution given a fixed stimulation area. In this section, we use model cells described by (4) with ($Q = 1$, $M = 9$, $P = 9$) to demonstrate real-time performance.

In Table II, the real-time performance is measured by the number of model cells that can be processed in real-time. The performance testing is done under conditions of different temporal and spatial resolutions. The temporal resolutions is given by a set of pattern rates (1440Hz, 720Hz, 360Hz) and the corresponding pulse durations (the reciprocal of the pattern rate: 0.7ms, 1.4ms, 2.8ms). The spatial resolution is given by the size of super-pixels in terms of visual angle (0.08°, 0.12°, 0.16°).

Since the real-time constraint requires the system to finish processing one frame before the next one arrives, the scalability is limited by the frame rate F_s. Given a lower frame rate, more model cells can be processed within a frame period T_s. While a lower frame rate increases the scalability, it decreases the temporal resolution of optogenetic stimulation as the temporal resolution is determined by the pattern rate F_p,

TABLE II
THE NUMBER OF MODEL CELLS THAT CAN BE PROCESSED IN REAL-TIME AND THE CORRESPONDING SPATIAL RANGE (IN PARENTHESES) UNDER CONDITIONS OF DIFFERENT TEMPORAL AND SPATIAL RESOLUTIONS

Spatial resolution \ Temporal resolution	$F_p = 1440Hz, t_p = 0.7ms$	$F_p = 720Hz, t_p = 1.4ms$	$F_p = 360Hz, t_p = 2.8ms$
$\triangle_{out,2} = 0.08°$	3.5K (5.3°)	7.4K (7.8°)	15.0K (11.1°)
$\triangle_{out,3} = 0.12°$	3.5K (8.0°)	7.3K (11.6°)	14.5K (16.3°)
$\triangle_{out,4} = 0.16°$	2.6K (9.2°)	7.0K (15.1°)	14.3K (21.6°)

which equals 24 times the frame rate F_s. The tradeoff between scalability and temporal resolution is shown in Table II: the number of model cells that can be processed in real-time is inversely proportional to the temporal resolution, which is about 3K, 7K, and 14K with a temporal resolution of 1440Hz, 720Hz, and 360Hz, respectively.

As shown in Table II, the scalability is also affected by the spatial resolution but to a much lesser degree. In each column (where the temporal resolution is fixed), as the size of the super-pixel increases, the number of model cells processed in real time remains relatively constant. As a result, the spatial range of optogenetic stimulation is almost proportional to the size of the super-pixel (and thus inversely proportional to the spatial resolution). This indicates that another tradeoff exists between spatial resolution and spatial range: given a fixed temporal resolution, a lower spatial resolution enables a larger stimulation area.

While remaining relatively constant, in each column, there is a decrease in the number of model cells as the spatial range of optogenetic stimulation increases. This is because a larger stimulation area requires a larger visual field (and thus a larger image) to be processed. Note that all the decreases are slight with one exception: there is a drop from 3.5K to 2.6K in the case of 60Hz frame rate. This is because when processing larger images, the GPP requires more time to process and transfer data so that the DSP cannot be taken maximum use. The overhead could be critical for a short frame period (16.7ms in this case).

As we mentioned before, there is a division of work between the GPP and the DSP: the GPP is responsible for preprocessing and transferring data and the DSP is reserved for intensive neural computing. While the real-time performance is typically limited by the DSP, this case shows that the GPP could also become the limiting factor as the size of image increases.

VI. DISCUSSION

In this section, we gives a detailed discussion on the applicability and limitations.

A. Platform

We have proposed a DMD based platform for optogenetic applications in sensory restoration. Another application of optogenetics is neurological diseases such as Parkinson's disease and epilepsy. In these diseases, the devices work largely by stimulating or inhibiting many neurons in a region (e.g., to balance neuronal signaling); this is, at least in part, because neuronal models for how these areas work and what causes to malfunction is still under study, so the increased value of building high spatial resolution is not yet known.

Optogenetic applications in these diseases are still very much in the research stage. Currently, the goal is to dissect the circuitry and determine points at which intervention is possible within animal models. In epilepsy, optogenetics has been applied to abort seizure-like activities [8], [9]. To achieve a closed-loop control, the processor uses EEG signal as the sensing input, and performs real-time seizure detection. Upon detection, light is delivered to suppress excessive neuronal activity in targeted brain regions (such as hippocampus and thalamus). In Parkinson's disease, optogenetics has been used to define specific nuclei and projections for the therapeutic effect of deep brain stimulation [10], [11]. Closed-loop control is not necessary for these studies.

In both cases, the targeted brain areas are located deep in the brain. While DMDs can offer high spatial resolution, they are only applicable to 2-dimensional superficial surfaces, such as the retina or the surface of the cortex. To stimulate deep brain structures, optogenetic stimulation is typically performed with implanted optical fibers and the light is supplied by an external source such as high power LEDs [32]. The implanted optical fibers guide light to regions of interest. In comparison with DMDs, the fiber-coupled LED approaches are limited by the spatial resolution they can offer. Nevertheless, the limitation has been lessened by the demonstrated feasibility of increasing the number of optical stimulation channels [33].

In the proposed platform, a single board computer (Overo Water) was adopted as the embedded system module. The choice is mostly due to its convenience as it provides all the peripheral components and interfaces necessary for video processing and DLP control. This makes development easier, faster, and less expensive. The platform is appealing as it is readily usable for a range of applications. Nevertheless, for designs that demand more processing power, customized circuit boards may need to be built to accommodate high-end processors or FPGAs to increase the processing capacities.

B. Model

The focus of the paper is on the implementation aspects of neuronal models in an embedded platform for optogenetic prosthetic applications. We have demonstrated general principles of implementation in such a platform to meet the requirements of a range of applications. Firing-rate models were proposed as they provide a simpler description of neural dynamics and can be simulated rapidly.

In this paper, the firing rate model in the form of (4) and (7) was employed as an example due to its generic nature. The model consists of a linear difference equation (which specifies an infinite impulse response function as the kernel) for spatial and temporal integration, and a nonlinear function for phenomena such as spike threshold and response saturation.

The use of the model to describe neural dynamics has been established and validated at the cellular and network levels [34]–[39]. It has also been adapted to various applications in nervous systems. In some applications, the model may evolve into variant forms but the general structure persists and the same implementation strategies apply. For example, when applied to describe stimulus-response characteristics in sensory pathways such as retina [20], [40], [41], lateral geniculate nucleus (LGN) [42], and visual cortex [43], the kernel is more conveniently specified by a finite impulse response function instead of a difference equation (the two forms are convertible from one to the other [44]).

Since the paper focuses on implementation aspects, a generic model was employed for the purpose of demonstration. Readers interested in model development are pointed to examples at various levels of nervous system [34]–[43]. Especially, those who are interested in prosthetic applications at the retinal level are referred to our study [20], in which models were developed for retinal prosthetics and the performance was demonstrated by cross-validating it with data from in-vitro retinal recording.

VII. Conclusion

In this paper, we have proposed an embedded processing platform for optogenetic stimulation. The platform consists of an embedded system module, and a portable digital light processing projector. As an optogenetic stimulator, the projector is capable of delivering light with high intensity as well as high spatial and temporal resolution. To mimic the processing of nervous system in real time, a heterogeneous GPP-DSP architecture of the embedded system module has been explored to implement compact model cells. An illustrative example is given to illustrate design principles from both software and hardware angles, and the real-time performance of the processing platform has been demonstrated. Constructed with off-the-shelf components, the high-efficiency and low-cost (less than $1000) platform can be easily adapted, upgraded, and customized to meet the needs of a wide range of biomedical applications.

References

[1] G. Nagel *et al.*, "Channelrhodopsin-2, a directly light-gated cation-selective membrane channel," *Proc. Nat. Acad. Sci. USA*, vol. 100, no. 24, pp. 13940–13945, 2003.

[2] F. Zhang *et al.*, "Multimodal fast optical interrogation of neural circuitry," *Nature*, vol. 446, no. 7136, pp. 633–639, 2007.

[3] X. Han *et al.*, "Millisecond-timescale optical control of neural dynamics in the nonhuman primate brain," *Neuron*, vol. 62, no. 2, pp. 191–198, 2009.

[4] L. Petreanu, D. Huber, A. Sobczyk, and K. Svoboda, "Channelrhodopsin-2-assisted circuit mapping of long-range callosal projections," *Nature Neurosci.*, vol. 10, no. 5, pp. 663–668, 2007.

[5] O. G. S. Ayling, T. C. Harrison, J. D. Boyd, A. Goroshkov, and T. H. Murphy, "Automated light-based mapping of motor cortex by photoactivation of channelrhodopsin-2 transgenic mice," *Nature Methods*, vol. 6, no. 3, pp. 219–224, 2009.

[6] K. M. Tye and K. Deisseroth, "Optogenetic investigation of neural circuits underlying brain disease in animal models," *Nature Rev. Neurosci.*, vol. 13, no. 4, pp. 251–266, 2012.

[7] J. Tønnesen, A. T. Sørensen, K. Deisseroth, C. Lundberg, and M. Kokaia, "Optogenetic control of epileptiform activity," *Proc. Nat. Acad. Sci. USA*, vol. 106, no. 29, pp. 12162–12167, 2009.

[8] J. T. Paz *et al.*, "Closed-loop optogenetic control of thalamus as a tool for interrupting seizures after cortical injury," *Nature Neurosci.*, vol. 16, pp. 64–70, Nov. 2012.

[9] E. Krook-Magnuson, C. Armstrong, M. Oijala, and I. Soltesz, "On-demand optogenetic control of spontaneous seizures in temporal lobe epilepsy," *Nature Commun.*, vol. 4, no. 1376, pp. 1–8, 2013.

[10] V. Gradinaru, M. Mogri, K. R. Thompson, J. M. Henderson, and K. Deisseroth, "Optical deconstruction of parkinsonian neural circuitry," *Science*, vol. 324, pp. 354–359, Apr. 2009.

[11] A. V. Kravitz *et al.*, "Regulation of parkinsonian motor behaviors by optogenetic control of basal ganglia circuitry," *Nature*, vol. 466, pp. 622–626, Jul. 2010.

[12] V. H. Hernandez *et al.*, "Optogenetic stimulation of the auditory pathway," *J. Clin. Invest.*, vol. 124, no. 3, pp. 1114–1129, 2014.

[13] A. Bi *et al.*, "Ectopic expression of a microbial-type rhodopsin restores visual response in mice with photoreceptor degeneration," *Neuron*, vol. 50, pp. 23–33, 2006.

[14] P. S. Lagali *et al.*, "Light-activated channels targeted to ON bipolar cells restore visual function in retinal degeneration," *Nature Neurosci.*, vol. 11, pp. 667–675, Apr. 2008.

[15] Y. Zhang, E. Ivanova, A. Bi, and Z.-H. Pan, "Ectopic expression of multiple microbial rhodopsins restores ON and OFF light responses in retinas with photoreceptor degeneration," *J. Neurosci.*, vol. 29, no. 29, pp. 9186–9196, 2009.

[16] N. Grossman *et al.*, "Multi-site optical excitation using ChR2 and micro-LED array," *J. Neural Eng.*, vol. 7, no. 1, p. 16004, 2010.

[17] H. Tomita *et al.*, "Channelrhodopsin-2 gene transduced into retinal ganglion cells restores functional vision in genetically blind rats," *Experim. Eye Res.*, vol. 90, no. 3, pp. 429–436, 2010.

[18] S. Thyagarajan, M. van Wyk, K. Lehmann, S. Löwel, G. Feng, and H. Wässle, "Visual function in mice with photoreceptor degeneration and transgenic expression of channelrhodopsin 2 in ganglion cells," *J. Neurosci.*, vol. 30, no. 26, pp. 8745–8758, 2010.

[19] M. M. Doroudchi *et al.*, "Virally delivered channelrhodopsin-2 safely and effectively restores visual function in multiple mouse models of blindness," *Molecular Therapy*, vol. 19, no. 7, pp. 1220–1229, 2011.

[20] S. Nirenberg and C. Pandarinath, "Retinal prosthetic strategy with the capacity to restore normal vision," *Proc Nat. Acad. Sci. USA*, vol. 109, no. 37, pp. 15012–15017, 2012.

[21] C. Pandarinath, E. T. Carlson, and S. Nirenberg, "A system for optically controlling neural circuits with very high spatial and temporal resolution," in *Proc. IEEE 13th Int. Conf. Bioinf. Bioeng. (BIBE)*, Nov. 2013, pp. 1–6.

[22] B. Yan, M. Vakulenko, S. H. Min, W. W. Hauswirth, and S. Nirenberg, "Maintaining ocular safety with light exposure, focusing on devices for optogenetic stimulation," *Vis. Res.*, vol. 24, no. 121, pp. 57–71, 2016.

[23] S. Kleinlogel *et al.*, "Ultra light-sensitive and fast neuronal activation with the Ca^{2+}-permeable channelrhodopsin CatCh," *Nature Neurosci.*, vol. 14, no. 4, pp. 513–518, Mar. 2011.

[24] N. C. Klapoetke *et al.*, "Independent optical excitation of distinct neural populations," *Nature Methods*, vol. 11, no. 3, pp. 338–346, Feb. 2014.

[25] D. Dudley, W. M. Duncan, and J. Slaughter, "Emerging digital micromirror device (DMD) applications," *Proc. SPIE*, vol. 4985, pp. 14–25, Jan. 2003.

[26] B. Yan and P. Li, "Reduced order modeling of passive and quasi-active dendrites for nervous system simulation," *J. Comput. Neurosci.*, vol. 31, no. 2, pp. 247–271, 2011.

[27] M. Wang, B. Yan, J. Hu, and P. Li, "Simulation of large neuronal networks with biophysically accurate models on graphics processors," in *Proc. IEEE Int. Joint Conf. Neural Netw.*, Jul. 2011, pp. 3184–3193.

[28] Y. Zhang, B. Yan, M. Wang, J. Hu, H. Lu, and P. Li, "Linking brain behavior to underlying cellular mechanisms via large-scale brain modeling and simulation," *Neurocomputing*, vol. 97, no. 15, pp. 317–331, 2012.

[29] P. Dayan and L. F. Abbott, *Theoretical Neuroscience: Computational and Mathematical Modeling of Neural System*. Cambridge, MA, USA: MIT Press, 2001.
[30] A. V. Cideciyan *et al.*, "Developing an outcome measure with high luminance for optogenetics treatment of severe retinal degenerations and for gene therapy of cone diseases," *Invest. Ophthalmol. Vis. Sci.*, vol. 57, no. 7, pp. 3211–3221, 2016.
[31] N. V. Swindale, "Visual map," *Scholarpedia*, vol. 3, no. 6, pp. 4607, 2008.
[32] R. Pashaie *et al.*, "Optogenetic brain interfaces," *IEEE Rev. Biomed. Eng.*, vol. 7, pp. 3–29, 2014.
[33] G. Gagnon-Turcotte, Y. LeChasseur, C. Bories, Y. Messaddeq, Y. D. Koninck, and B. Gosselin, "A wireless headstage for combined optogenetics and multichannel electrophysiological recording," *IEEE Trans. Biomed. Circuits Syst.*, vol. 11, no. 1, pp. 1–14, Jan. 2017.
[34] H. R. Wilson and J. D. Cowan, "Excitatory and inhibitory interactions in localized populations of model neurons," *Biophys. J.*, vol. 12, pp. 1–24, Jan. 1972.
[35] B. Ermentrout, "Reduction of conductance-based models with slow synapses to neural nets," *Neural Comput.*, vol. 6, no. 4, pp. 679–695, 1994.
[36] W. Gerstner, "Time structure of the activity in neural network models," *Phys. Rev. E, Stat. Phys. Plasmas Fluids Relat. Interdiscip. Top.*, vol. 51, pp. 738–758, Jan. 1995.
[37] O. Shriki, D. Hansel, and H. Sompolinsky, "Rate models for conductance-based cortical neuronal networks," *Neural Comput.*, vol. 15, no. 8, pp. 1809–1841, 2003.
[38] S. Ostojic and N. Brunel, "From spiking neuron models to linear-nonlinear models," *PLoS Comput. Biol.*, vol. 7, pp. 1–16, Jan. 2011.
[39] E. S. Schaffer, S. Ostojic, and L. F. Abbott, "A complex-valued firing-rate model that approximates the dynamics of spiking networks," *PLoS Comput. Biol.*, vol. 9, pp. 1–11, Oct. 2013.
[40] V. J. Uzzell and E. J. Chichilnisky, "Precision of spike trains in primate retinal ganglion cells," *J. Neurophysiol.*, vol. 92, no. 2, pp. 780–789, 2004.
[41] J. W. Pillow *et al.*, "Spatio-temporal correlations and visual signalling in a complete neuronal population," *Nature*, vol. 454, pp. 995–999, Aug. 2008.
[42] R. C. Liu, S. Tzonev, S. Rebrik, and K. D. Miler, "Variability and information in a neural code of the cat lateral geniculate nucleus," *J. Neurophysiol.*, vol. 86, no. 6, pp. 2789–2806, 2001.
[43] C. L. Baker, "Linear filtering and nonlinear interactions in direction-selective visual cortex neurons: A noise correlation analysis," *Vis. Neurosci.*, vol. 18, no. 3, pp. 465–485, 2001.
[44] B. Beliczynski, I. Kale, and G. D. Cain, "Approximation of FIR by IIR digital filters: An algorithm based on balanced model reduction," *IEEE Trans. Signal Process.*, vol. 40, no. 3, pp. 532–542, Mar. 2017.