33rd Annual International Conference of the IEEE EMBS
Boston, Massachusetts USA, August 30 - September 3, 2011

ARM-Based Visual Processing System for Prosthetic Vision

Paul. B. Matteucci, *Member, IEEE*, Philip Byrnes–Preston, Spencer C. Chen, *Member, IEEE,* Nigel H. Lovell, *Fellow, IEEE* and Gregg J. Suaning, *Member, IEEE*

***Abstract*—A growing number of prosthetic devices have been shown to provide visual perception to the profoundly blind through electrical neural stimulation. These first-generation devices offer promising outcomes to those affected by degenerative disorders such as retinitis pigmentosa. Although prosthetic approaches vary in their placement of the stimulating array (visual cortex, optic-nerve, epi-retinal surface, sub-retinal surface, supra-choroidal space, etc.), most of the solutions incorporate an externally-worn device to acquire and process video to provide the implant with instructions on how to deliver electrical stimulation to the patient, in order to elicit phosphenized vision. With the significant increase in availability and performance of low power-consumption smart phone and personal device processors, the authors investigated the use of a commercially available ARM (Advanced RISC Machine) device as an externally-worn processing unit for a prosthetic neural stimulator for the retina. A 400 MHz Samsung S3C2440A ARM920T single-board computer was programmed to extract 98 values from a 1.3 Megapixel OV9650 CMOS camera using impulse, regional averaging and Gaussian sampling algorithms. Power consumption and speed of video processing were compared to results obtained to similar reported devices. The results show that by using code optimization, the system is capable of driving a 98 channel implantable device for the restoration of visual percepts to the blind.**

I. Introduction

Visual percepts have been elicited in vision impaired patients though the electrical stimulation of the visual cortex [1, 2], the optic nerve [3] and the eye [4, 5]. Although on-implant image processing has been shown to be possible [6], various groups [7, 8] have opted to use some form of image acquisition device and associated computation circuitry in order to process the visual scene. Importantly, this approach provides researchers with the opportunity to improve and upgrade the capabilities of the implant system without the need for revision surgery to modify implanted components.

External processing of the visual scene ultimately leads to a set of stimulation instructions that are sent to the implanted hardware, typically by the way of an inductively-coupled transcutaneous link. Reports on present generation visual neuroprostheses, irrespective of their electrode quantities, elicit phosphene percepts in small numbers and at spatial separations consistent with very rudimentary vision.

In the case of the device being developed by the authors, images can be acquired from a camera or manually uploaded to the portable processing device; the images are sampled and transmitted through an inductive radio frequency (RF) link to a device with 98 electrodes surgically implanted within the suprachoroidal space of the eye (Fig. 1).

The aim was to determine whether a low-cost, commercially-available, portable device could prove to be sufficiently powerful to perform the image processing tasks required to drive a 98 electrode visual prosthesis with meaningful data that results in useful visual information. Important considerations for the system also include being sufficiently lightweight to be worn by a patient for extended periods of time and with a current draw which allows for it to be powered by a compact battery for several hours.

A key, limiting factor in previously reported devices [8, 9] is the relatively slow data transfer rate between the system's processor and the camera. In order to provide a fluid image to the prosthetic device, the system should provide a sufficiently fast connection to refresh the visual scene at a rate of at least 20 Hz so as to reduce the propensity of the visual scene to appear as "flickering" to the patient.

In this paper the authors present an ARM-based processing device capable of performing real time image processing and transmitting resulting stimulations parameters via a serial connection. Particular emphasis was given to the optimization of the software in order to achieve real time processing. A Technical Interface (TI) software was written to simulate the tools a physician may require to test and configure the device once implanted in a patient.

II. Methodology

A. Hardware

A Mini2440 single-board computer was used (FriendlyARM Computer Technology, Guangzhou, China)(Fig. 2); it incorporates a 400 MHz Samsung S3C2440 ARM9 processor with 64MB RAM and 1GB of solid state storage. The Mini2440 was chosen due to its processor speed, compact size (100 * 100mm), light weight (176g), low cost, open-source framework and the presence of a dedicated CMOS camera port on the chip that allowed the acquisition of images from the camera at a much higher frame rate than that of typical USB interfaces. An OV9650 1.3 Megapixels CMOS camera (Omnivision, California, USA) was used; this was chosen for its hardware compatibility, driver availability and the possibility of streaming video at 60 fps at 320 * 240 pixel resolution. Due to Linux driver limitations, the camera was run at 50fps. In order to test battery life a 6 Ah three-cell lithium polymer

Matteucci, Byrnes-Preston, Chen, Lovell and Suaning: Graduate School of Biomedical Engineering, University of New South Wales, Sydney, Australia.

battery (3.7 V nominal, 2000 mAh per cell) was sourced

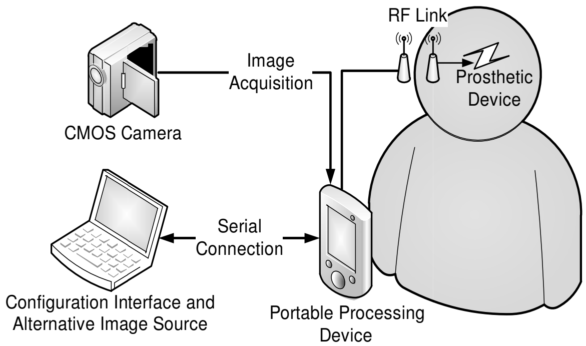

Fig. 1. Interaction amongst the image processing components. The external camera acquires images which are processed by the portable device and sent to the prosthetic device by means of the RF link. A monitor computer can be used to control sampling parameters.

(cell 585460, Unionfortune Electronics, Guangdong, China) and connected to a 5 V boost regulator.

B. Portable Device Software

The operating system chosen was Linux (kernel 2.6.32). The camera driver was modified to reduce the resolution from 640 * 480 to 320 * 240 pixels whilst increasing the frame rate from 25 fps to 50 fps.

The sampling program was written in C++ and cross-compiled using ARM-GCC (4.3.3) on a development desktop PC running Ubuntu Linux (10.10). Particular emphasis was given to structuring the program into a sequence of logical steps to allow collaborating research groups to easily implement their own algorithms and test their relative efficacy.

Upon start-up of the device, the program would automatically launch and perform the following:

1. Read (or generate a default setting) a configuration file containing hardware and sampling parameters;
2. Initialization of the hardware enabled in the configuration file (camera, screen and serial interfaces);
3. Check for serial commands from the TI;
4. Acquire a frame from the camera;
5. Pre-process the frame (light normalization);
6. Sample the image;
7. Output stimulation parameters to the test RF link and TI.

Unless a fixed number of frames was specified in the configuration file, the program ran continuously.

Based on previous image processing research [10], three sampling algorithms were implemented:

- Impulse Sampling (IS);
- Regional Averaging (RA);
- Gaussian Sampling (GS).

The adopted sampling algorithm was defined in the configuration file, but could be changed through the TI if required.

For IS, the electrode position was mapped to the 320 * 240 image from the camera buffer, and the 16 bit RGB pixel value, was obtained. The pixel's 16 bit RGB value was then converted to a value for luminance and used as the value for amplitude of the biphasic stimulus waveform and output to the serial port. RA and GS utilized the same technique to convert pixel luminance to biphasic stimulus, however for RA a square area was mathematically averaged, and for GS a 2D circular symmetrical Gaussian distribution was used to give more weight to the center pixel while still encompassing information from the surrounding pixels. Sampling area is selectable from the TI or in the configuration file of the sampling program.

The conversion from serial to RF and error checking is intended to be implemented by a separate and dedicated device.

C. External Configuration Software

The TI was written in C# (.NET 4.0) and run on Windows (XP or 7). The TI was designed as a tool to configure a device to the specific requirements of a patient, to account for biological variation, differences in electrode-tissue interfaces and extent of disease progression.

A wired serial connection (RS-232 or Serial to USB converter) was used between the TI and the portable device for communication.

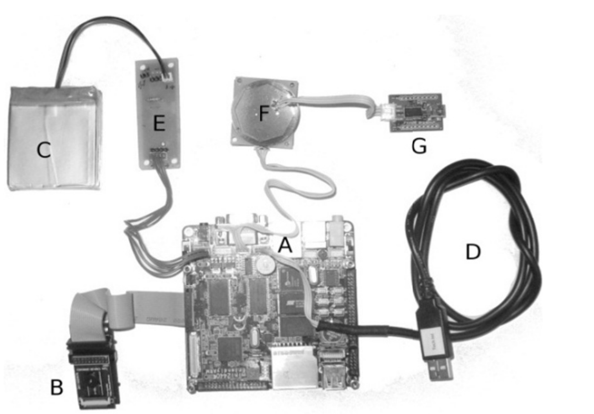

Fig. 2. (A) Main components of the portable device: single board computer, (B) CMOS camera, (C) 6Ah lithium polymer battery, (D) USB connection to TI, (E) battery monitoring hardware, (F) test RF transmitter, (G) RF serial to USB output converter for testing.

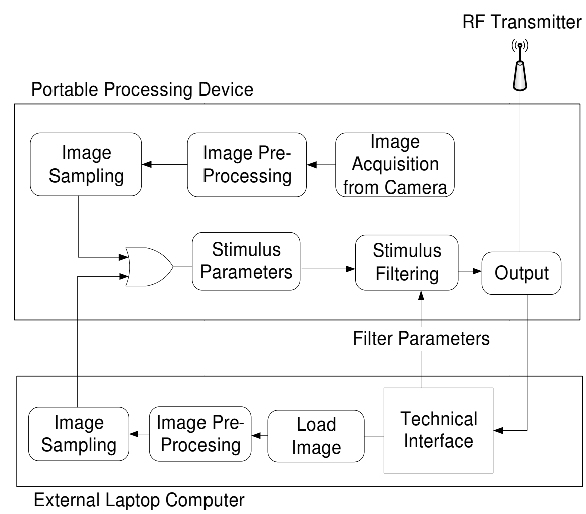

Fig. 3. Outline of the process used to convert streaming video (from the camera) or static images (from the laptop computer) into stimulus parameters, and sent to the implanted device.

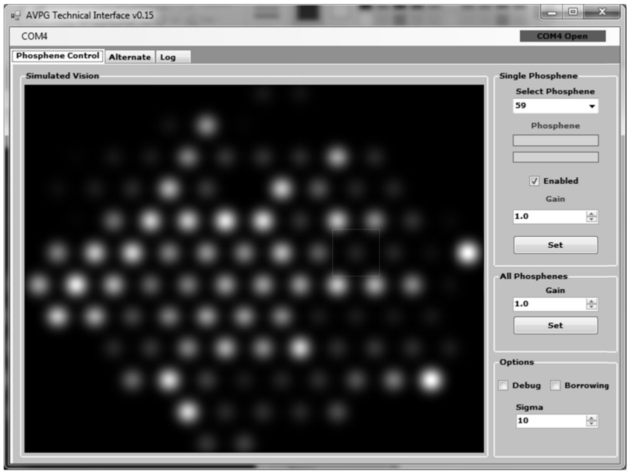

Fig. 4. The Technical Interface's Phosphene Control panel. A phosphenised representation of a hand is shown in the main panel. The right panel presents various parameters which can be used to modify the output of the portable device to the prosthesis.

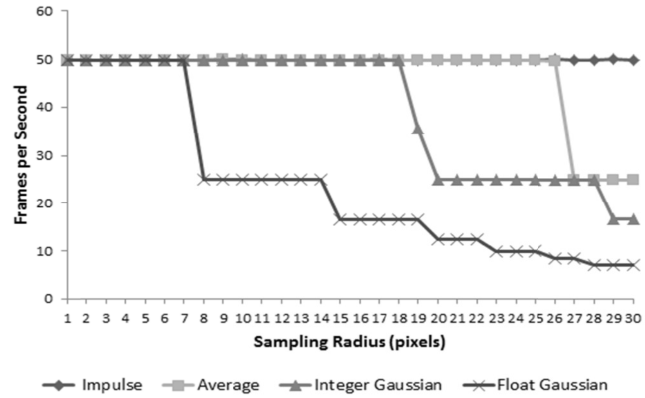

Fig. 5. Speed of different sampling techniques. Gaussian and average sampling speeds decrease with the increase in sampled area, however a speed optimization allowed for increase in sampling area with a smaller impact in performance.

Two main panels were available to the user of the TI: Phosphene Control (PC) panel (Fig. 4) was designed to tailor the neurostimulation parameters to the specific requirements of the patient; the Alternate Panel (AP) allowed the portable device to bypass the camera and output to the implanted device a fixed set of stimulation instructions generated by the TI from a pre-loaded image. The visual representation of the PC panel did not represent the actual imperfections of the perception of vision elicited in a patient (i.e. variation in luminance, phosphene size, shape and position). However the functions allowed for the manipulation of individual phosphenes to more closely approximate those perceived by the patient. Further phosphene customization parameters will be implemented (variable phosphene position and shape).

To modify the parameters, individual phosphenes had to be selected; once selected, phosphene representations could be disabled if no phosphene was perceived from the corresponding electrode, or its gain manipulated in order to compensate for changes in perceptual thresholds. Changes to the thresholds were transmitted to the portable device where they were stored as a matrix. Before being output to the RF transmitter, the amplitude of the stimulus of each electrode was multiplied by the gain matrix.

The initial testing of an implanted device will require testing of individual electrodes and the testing of visual perception using static images. The AP will allow the assisting physician to fix the output of the portable device to an arbitrary image. This can be used for psychophysics testing and training to recognize patterns and images.

III. RESULTS

A. Sampling Speed

Software developed for psychophysical experimentation by the authors' group [10] was initially loaded onto the portable device but the speed was found to be insufficient to satisfy the requirements for real time image processing, with the resulting frame rate ranging between 14 and 7 frames per second, depending on the sampling parameters (RA or GS). This was due to the sampling program being developed to run on a powerful desktop PC, which required large numbers of floating point calculations to be performed, and consequentially no specific optimization for portable devices. The sampling algorithm was rewritten with particular emphasis on real time speed of execution. In order to achieve this, the time necessary to acquire, pre-process and sample frames would have to be sufficiently small as to occur in between frame acquisitions of the camera. If the time taken to process a frame was in excess of the 20 ms between camera frame acquisitions, the camera would begin acquiring the subsequent frame causing the program to "skip" a frame resulting in a halving of the frame rate (Fig. 5); if the time exceeded 30 ms, two out of three frames would be skipped, reducing the frame-rate to 16 fps, etc.

Although the position of the electrodes on the implantable array previously described [11] had to be hard-coded, the corresponding sampling coordinates for the camera frame were mapped at load time. For the purposes of testing for this paper, a fixed mapping was used; however this allows subsequent designs to implement adjustable electrode mappings to compensate for perceptual distortion in degenerate retinas, with no additional computational delay.

The architecture of the S3C2440 does not include a floating point co-processor, which means that decimal point operations require multiple clock cycles to be executed. In order to offset this limitation, the pre-calculated Gaussian matrices were converted to integers by multiplying each value by a fixed multiplier (1048576, equal to a 20 bit shift). Once the Gaussian sampling was completed, the resulting value was then divided by the multiplied value to obtain the actual sampling value. This significantly increased the performance of the Gaussian sampling (Fig. 5) and allowed for real time Gaussian sampling for radii up to 18 pixels.

Further speed enhancements were obtained by pre-calculating all of the Gaussian matrices at load time and before the actual video stream sampling began.

B. Sampling Area

When presenting visual information to a patient using a first-generation device, it is important to maximize the information being conveyed in the few phosphenes available. An increase of sampling area and complexity of

TABLE I
MEAN AND VARIANCE OF POWER CONSUMPTION FOR DIFFERENT SAMPLING TECHNIQUES AVERAGED OVER 5 RUNS OF 60S

Sampling Type	CPU Activity (%)	Current Draw with screen (mA)	Current Draw no screen (mA)
Idle	0 ± 1%	576 ± 2	246 ± 1
IS	51.4 ± 0.6%	714 ± 2	392 ± 2
RA (r = 15px)	68.8 ± 0.5%	736 ± 1	409 ± 3
GS (r = 15px)	86.4 ± 0.2%	746 ± 1	416 ± 1
GS (r = 18px)	94.0 ± 0.2%	762 ± 2	432 ± 1

the sampling algorithm require an increase in computing speed and power consumption (Table 1).

In subsequent tests, 18 pixel GS was used. This value was chosen because it maximized the sampling radius without causing the drop in frame-rate. In the Gaussian equation, given a value of σ of 6 (¼ of the sampling point distance of 26 pixels), a radius of 18 allowed sampling of values with weights as low as 4.9e-5. Moreover, the sampling radii overlap, resulting in a sampling area which covers 75.5% of the total image acquired by the camera, hence effectively compressing a large amount of visual information.

C. Power Consumption

The true RMS AC+DC current draw of the device was monitored using a Tenma 72-7730 multi-meter (Tenma Test Equipment, Springboro, USA) whilst idle and running various sampling algorithms. The main sources of power consumption were determined to be the screen matrix and backlight, and the CPU activity (table I). The device was then connected to the lithium ion battery and the battery life tested. The linear regulator maintained 5V output until the battery voltage dropped below 2.7V, resulting in a battery life of 5 hours. The screen was disconnected and the tests repeated, resulting in a significant increase in battery life (6.8h ± 0.2h). It should be noticed that the RF link used in these tests is not used to power an implant, and therefore draws significantly less current than the final device.

IV. Discussion and Conclusion

In this paper the authors tested the possibility of using a low-cost, commercially available ARM-based single-board computer for video acquisition and processing for a vision prosthesis device. By utilizing platform specific algorithms, a modified camera driver and optimizing the hardware for power conservation, it was possible to develop a flexible, lightweight, portable device capable of sampling video from a camera in real time, which could be tailored to the individual threshold requirements of a patient.

In the last few years various techniques have been proposed to perform image processing and extrapolate salient information from a video stream for visual neuroprostheses [9, 12, 13], however, when it comes to hardware, it is often difficult to compare devices due to the time passing between reports and the nature of the device being developed. Fink et al. [9] proposed a much lighter (8 g) device which ran at a higher clock speed (600 MHz), however the compact nature of the hardware used excluded the presence of a dedicated camera port, resulting in a reduction in resolution to 160 * 120 pixels in order to achieve a satisfactory frame-rate. Moreover, the powerful CPU was mostly under-utilized, yielding a load of just 10%.

FPGAs also provide a viable hardware solution: Srivastava et al. [14] presented a high resolution, high phosphene-count FPGA to drive a visual implant, however implementing such a processing device on FPGA introduces rigid constraints for hardware upgrades; by adopting an ARM based device running Linux, the optimized image sampling software can be run on any device with compatible hardware. It is interesting to note how development of ARM processors for the smartphone and low-power markets have contributed to the biomedical field: Tsai et al. [8] in 2009, reported an image processing device capable of sampling a USB camera at a resolution of 176 * 144 at 30 fps, but requiring a DSP to perform the image processing.

With such rapid increases in performance, future devices will allow higher resolution image acquisition and the implementation more mathematically demanding algorithms.

References

[1] W. H. Dobelle, "Artificial vision for the blind by connecting a television camera to the visual cortex," *ASAIO J,* vol. 46, pp. 3-9, Jan-Feb 2000.

[2] W. H. Dobelle, D. O. Quest, J. L. Antunes, T. S. Roberts, and J. P. Girvin, "Artificial vision for the blind by electrical stimulation of the visual cortex," *Neurosurgery,* vol. 5, pp. 521-7, Oct 1979.

[3] C. Veraart, C. Raftopoulos, J. T. Mortimer, J. Delbeke, D. Pins, G. Michaux, A. Vanlierde, S. Parrini, and M. C. Wanet-Defalque, "Visual sensations produced by optic nerve stimulation using an implanted self-sizing spiral cuff electrode," *Brain Research,* vol. 813, pp. 181-186, Nov 30 1998.

[4] E. Zrenner, "Will retinal implants restore vision?," *Science,* vol. 295, pp. 1022-5, Feb 8 2002.

[5] J. F. Rizzo and J. Wyatt, "Prospects for a visual prosthesis," *Neuroscientist,* vol. 3, pp. 251-262, Jul 1997.

[6] E. Zrenner, K. D. Miliczek, V. P. Gabel, H. G. Graf, E. Guenther, H. Haemmerle, B. Hoefflinger, K. Kohler, W. Nisch, M. Schubert, A. Stett, and S. Weiss, "The development of subretinal microphotodiodes for replacement of degenerated photoreceptors," *Ophthalmic Res,* vol. 29, pp. 269-80, 1997.

[7] N. J. Parikh, J. D. Weiland, M. S. Humayun, S. S. Shah, and G. S. Mohile, "DSP based image processing for retinal prosthesis," *Conf Proc IEEE Eng Med Biol Soc,* vol. 2, pp. 1475-8, 2004.

[8] D. Tsai, J. W. Morley, G. J. Suaning, and N. H. Lovell, "A wearable real-time image processor for a vision prosthesis," *Comput Methods Programs Biomed,* vol. 95, pp. 258-69, Sep 2009.

[9] W. Fink, C. X. You, and M. A. Tarbell, "Microcomputer-based artificial vision support system for real-time image processing for camera-driven visual prostheses," *Journal of Biomedical Optics,* vol. 15, pp. -, Jan-Feb 2010.

[10] S. C. Chen, G. J. Suaning, J. W. Morley, and N. H. Lovell, "Simulating prosthetic vision: I. Visual models of phosphenes," *Vision Res,* vol. 49, pp. 1493-506, Jun 2009.

[11] Y. T. Wong, N. Dommel, P. Preston, L. E. Hallum, T. Lehmann, N. H. Lovell, and G. J. Suaning, "Retinal neurostimulator for a multifocal vision prosthesis," *IEEE Trans Neural Syst Rehabil Eng,* vol. 15, pp. 425-34, Sep 2007.

[12] H. Guo, R. Qin, Y. Qiu, Y. Zhu, and S. Tong, "Configuration-based processing of phosphene pattern recognition for simulated prosthetic vision," *Artif Organs,* vol. 34, pp. 324-30, Apr 2010.

[13] N. Parikh, L. Itti, and J. Weiland, "Saliency-based image processing for retinal prostheses," *J Neural Eng,* vol. 7, p. 16006, Feb 2010.

[14] N. R. Srivastava, P. R. Troyk, and D. Bradley, "FPGA based visual prosthesis device for testing visual perception on non human primates," in *Electro/Information Technology, 2007 IEEE International Conference on*, 2007, pp. 21-25.