FPGA design for dual-spectrum Visual Scene Preparation in Retinal Prosthesis

Musa Al Yaman[1], Walid Al-Atabany[2], Alex Bystrov[1], Patrick Degenaar[1]
[1]*Dept. of Electrical and Electronic Engineering, Newcastle University, Newcastle upon Tyne, UK*
[2]*Dep. Of Biomedical Engineering, Helwan University, Egypt.*

***Abstract*— A method of Visual Scene Preparation for the patients suffering Retinitis Pigmentosa is implemented in hardware for the first time. The scene is captured with two cameras, one visible spectrum and one infra-red, in order to distinguish between the live and non-live objects. The live objects are subsequently emphasized in the output image, thus helping a patient to see the most significant detail with the healthy part of the retina. The implementation uses Verilog language and FPGA platform. A system prototype is analyzed and compared to MATLAB results.**

***Index Terms*—** Field Programmable Gate Arrays (FPGA), Anisotropic Diffusion Filter, Image Simplification, Real Time, Augmented Vision, Retinal Prosthesis, Infra-Red Camera.

I. Introduction

Retinal prosthesis considered the most hopeful way to restore vision for those with the blinding condition; Retinitis Pigmentosa (RP). This inherited, degenerative eye disease causes severe vision impairment leading to total blindness in some late stage patients [1]. The pathology of the condition is the malfunction of low-light sensing rod cells (predominant in the peripheral vision) leading to a progressive destruction of surrounding light sensing cone cells. However, while there is a loss of the light sensing cells and reorganization of the processing layers in the retina. The main signal transmission pathways and retinal ganglion cells are still largely intact. [2]. This has led to the development of retinal prostheses which have moved into the commercialization phase [3]. There are still many challenges in this field, and current returned vision is very rudimentary.

The discovery of the channelrhodopsin in 2003 has led to optogenetic/photonic stimulation approaches to retinal prosthesis [4]. In this case, opsin proteins are used to sensitize a new layer which can then be stimulated with short pulses of ultra high intensity light. [5], [6]. To this end, we have previously demonstrated this approach using an active matrix CMOS controlled Gallium Nitride LED array [7]. But even in this case we do not expect perfect visual return in the first instance. As such, image processing methods can be used to maximize the useful information to be transferred to the patient [8].

Although all forms of visual prosthesis necessitate the use of image processor, there has been little focus on integrated scene enhancement specifically for retinal prosthesis. Most previous enhancement methodologies have been for augmented reality approaches to improve the vision of those with partial visual loss. These have primarily focused on representing the scene in the form of segmented edges or by enhancing the contrast using histogram equalization. Our own efforts in this field have been to create an image processing platform to enhance the visual scene before downscaling and transmitting it into our micro LED stimulator arrays [9]. The platform was based on enhancing the scene contrast by cartoonizing and non-linearly downscaling it into a smaller size. However, to maximize the information of downscaled images, it is necessary to segment them into important and less important regions.

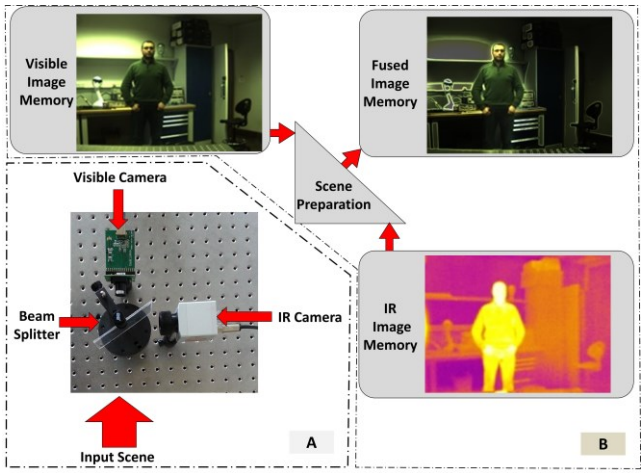

Figure 1 (A)The optical system alignement of the infrared and visible cameras, according to input scene. (B) General system flowchart illustrating the inputs and outputs of our Scene Preperation block.

Although many image segmentation techniques exist in the literature, we require one which can be implemented on a real time low power platform which can be operated on a battery. The work reported in this paper presents a novel FPGA-based implementation of an infrared camera as a based method to segmenting key features in the visual scene based on the object temperature. This multispectral technique is used to simplify the processing and thus processing energy cost. We demonstrate efficacy and low power real-time performance.

II. Methods

The original concept of infrared assisted scene segmentation was presented by Al-Atabany [10]. The system consists of two cameras; visible CMOS and infrared, which were optically aligned via a visible/IR beam splitter to view the same visual scene. In our experiments the two cameras are optically aligned as shown in Figure 1(A). The source images from the cameras and the resultant segmented image are shown in Figure 1(B). This is achieved by feeding the IR

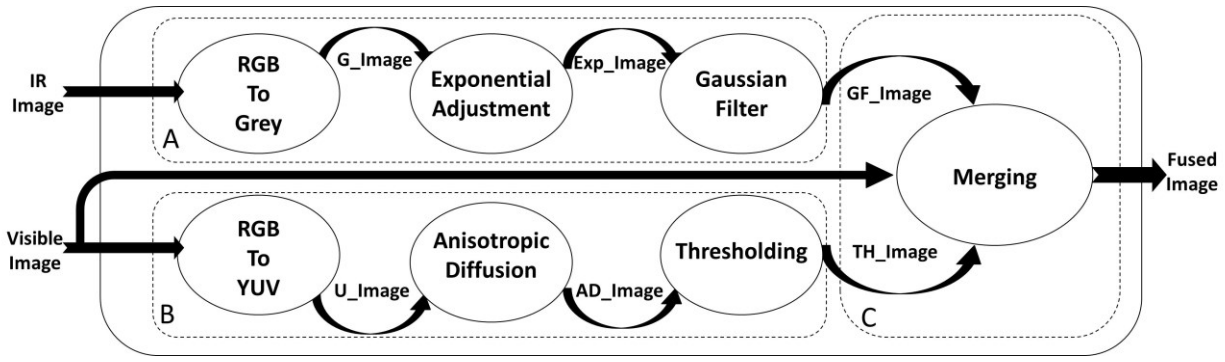

Figure 2 Full FPGA architecture blocks consist from **A.** the IR preparation 3 steps then **B.** the Visible preparation main three steps and finally **C.** the merging block that produces the final Fused Image.

and visible images into a scene preparation block on the FPGA from memory. The final fused image is transferred from the output buffer to output memory. Our scene preparation FPGA implementation is divided into three main parts labeled figure 2 and described in A, B and C below.

A. IR Preparation

The idea of using the infrared cameras comes from the fact that all objects act as black body radiators. They thus emit thermal photons of wavelength and intensity according their temperature. We, therefore, use this additional spectrum to segment objects with temperatures different than the ambient temperature. We start our IR preparation by converting the RGB input image (***IR Image***) into a Gray scale (***G_Image***).

An exponential scaling function is used to segment the hot and cold objects from the surrounding background. Exponential scaling changes the dynamic range of the image in order to enhance (boost) the high intensity pixel values while decreasing the low intensity pixel values. Thus, by utilizing such scaling on both low and high ends of the image, we can segment important structures at temperature extremes while ignoring the less important ambient temperatures at the median intensity of the image.

$$I_{\exp} = e^{0.025*I}$$
$$I_{Nexp} = e^{0.025*(255-I)} \quad (1)$$

Where $\boldsymbol{I}$ is the original image, $\boldsymbol{I_{exp}}$ and $\boldsymbol{I_{Nexp}}$ are the exponentially scaled images for the original and it's negative. Then the two exponentially scaled images are added together and scaled exponentially in order to suppress the low intensity pixel values. The resulted image includes the segmented cold and hot objects existed in the original infrared image.

$$IR_{segmented} = e^{0.025*(I_{\exp}+I_{Nexp})} \quad (2)$$

The results for the 256 possible intensities are calculated via a lookup table in our FPGA implementation. The outcome allows us to map the input image (***G_Image***) to output image (***Exp_Image***).

To remove any discontinuity, the segmented image is smoothed by convolving it with a Gaussian filter.

$$\widehat{IR}_{segmented} = IR_{segmented} * G(x,y)$$
$$G(x,y) = \frac{1}{2\pi\sigma^2} e^{-(x^2+y^2)/2\sigma^2} \quad (3)$$

In our FPGA implementation for the Gaussian filter, we have an input buffer for two incoming rows from the input image (***Exp_Image***) as discussed in details below, we use simple shift and addition for the calculations, as all coefficients of Gaussian filter are powers of 2. The final image in the IR preparation stage (***GF_Image***) is used as a decision map to segment and fuse the final enhanced image.

B. Visible Preparation

As for all current retina prosthesis devices, only a grayscale image will be used in stimulating the retina. Therefore, the visible RGB image (***Visible Image***) is firstly converted into YUV color space, which takes human perception into account, allowing reduced bandwidth for chrominance components, we concerned about the U value using the following equation to get the output image (***U_Image***):

$$\mathrm{U} = 0.25R + 0.5G + 0.09375B + 16 \quad (4)$$

Where $\boldsymbol{U}$ the 8 bit output grayscale image ***R, G*** and ***B*** are the 8 bit red, green and blue input colors of the image respectively. The input coefficients used in this equation are powers of 2, which mean that the internal FPGA implementation can be achieved through shifting and addition.

For purely visual segmentation, we use a cartoon enhancement approach. This improves the contrast of visually important features, by simplifying and reducing contrast in low-contrast regions and artificially increasing contrast in higher contrast regions. To achieve this cartoon effect we use anisotropic diffusion filtering which was initially suggested by Perona and Malik in 1990 [11]. This is a nonlinear iterative process which increasingly smoothes an image while preserving the significant edges. The equation of the anisotropic diffusion in the discrete domain is:

$$I^{n+1} = I^n + \Delta T[\nabla(C.\nabla I_H) + \nabla(C.\nabla I_V)] \quad (5)$$

Where $\boldsymbol{n}$ denotes the iteration number between 1 and N;$\nabla$is the gradient operator; $\boldsymbol{C}$ is the diffusion coefficient; $\Delta\boldsymbol{T}$ is the time step (it controls the accuracy and the speed of the smoothing) and $\boldsymbol{I_H}$, $\boldsymbol{I_V}$ represents the diffusion in horizontal and vertical directions.

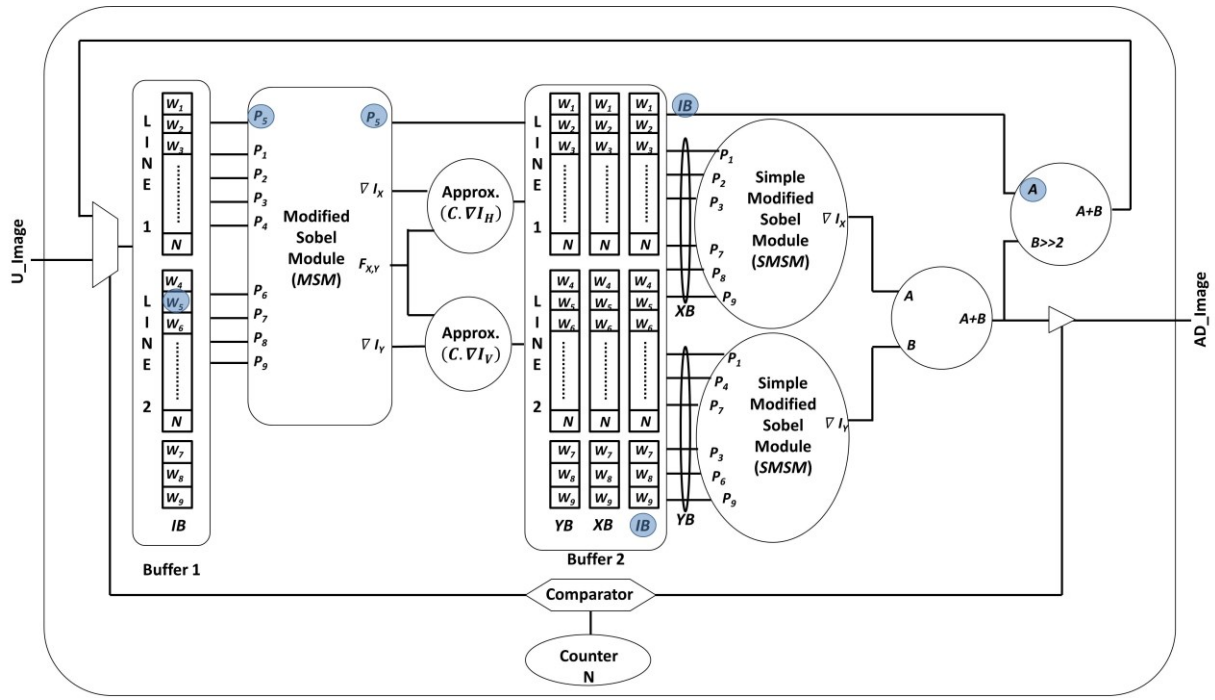

Figure 3 Full anisotropic diffusion filter architecture consists from Modified Sobel Module (MSM), two buffers with two Simple Modified Sobel Module (SMSM) that has the sme functionality of (MSM) but with eliminating unnecessery circuit elements, and finally simple add/shift circuits with approximation of the product terms ($C.\nabla I_V$) and ($C.\nabla I_H$).

The diffusion coefficient used in this paper is then calculated from the following equation:

$$C = \frac{1}{1 + \sqrt{\left(\nabla I_H{}^2 + \nabla I_V{}^2\right)}} \tag{6}$$

In figure 3 we present our anisotropic diffusion filtering. The information stream arrives from (***U_Image***) input to ***Buffer 1***, the buffer output drives the Modified Sobel Module (MSM) which has P_1, P_2, P_3, P_4, P_5, P_6, P_7, P_8 and P_9 to represent the nine 8-bit pixel inputs to the Modified Sobel Module. P_5 is the central pixel in the 3x3 convolution matrix. The module consists of simple signed subtractors, shift registers and adders. The MSM module has four outputs; the value of the pixel (P_5), the vertical and horizontal gradients∇I_X and ∇I_Y, and the absolute value of the combined gradients $F_{X,Y}$. $F_{X,Y}$ is calculated as follows, and used as an approximation in equation 7

$$F_{X,Y} = \sqrt{\left(\nabla I_H{}^2 + \nabla I_V{}^2\right)} \approx |\nabla I_H| + |\nabla I_V| \tag{7}$$

Finally the P_5 (I^n) output used to update the image in final step as equation 5 illustrates. To calculate diffusion coefficient ***(C),*** we need to add one to the ($F_{X,Y}$) output from the **MSM** and take the reciprocal for the addition result. Instead of this we divide the ∇I_X, ∇I_Y gradients by the summation result in order to get ($C.\nabla I_H$) and ($C.\nabla I_V$) respectively.

Buffer 2 consists of three sub-buffers**,** two of them to store the ($C.\ \nabla I_H$) and ($C.\ \nabla I_V$), and the third one used to store the pixel in the center of the convolution window. Simple Modified Sobel Module (SMSM) is used to compute the anisotropic diffusion result; the **SMSM** is similar to the **MSM** module except that we exclude some unused circuits to minimize power consumption. The output gradients from the **SMSM are then** added together and multiplied by **ΔT = 0.25** (shift left 2). The latter result is then added to the central pixel **P5**. Finally, the output is return back to input to do a next iteration (n), this is continues until the N counter (which holds the number of iterations) equals to zero then the final output image (***AD_Image***) will ready.

We then define two threshold values, τ_{min} ,τ_{max} and we set all pixels of the (***AD_Image***) below τ_{min} to 0 and all the pixels above τ_{max} are set to 255. The final image in the visible preparation stage (***TH_Image***) is used in fusing the final enhanced image.

C. Combining the IR and Visible Spectrum Images

The infrared segmented image (***GF_Image***) is used to create weighted decision regions by which a linear combination of the pixels in the visible (***Visible Image***) and the cartoon visible (***TH_Image***) images is used to generate corresponding pixels in the fused image. Then the fused image will be:

$$I_{fused(Cartoon)} = \widehat{IR}_{segmented} * I_{Visible(Cartoon)} + \left(1 - \widehat{IR}_{segmented}\right) * I_{Visible} \tag{8}$$

III. Results and Discussion

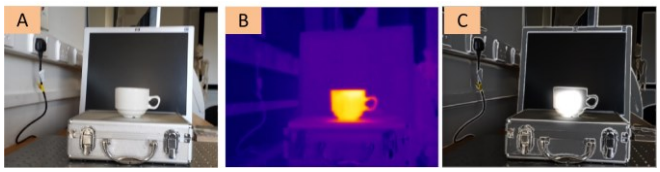

Figure 4 (A)The visible image of a cup contains hot water. (B) The Infrared image for the cup. (C) The fused image output.

In Figure 4 the two inputs of our algorithm are shown where (A) represents the visible image for the cup contains hot water, while in (B) the IR image shows the cup with orange color according to its high tempreture, the final fused image represented in (C) with the cup highligthed more than background.

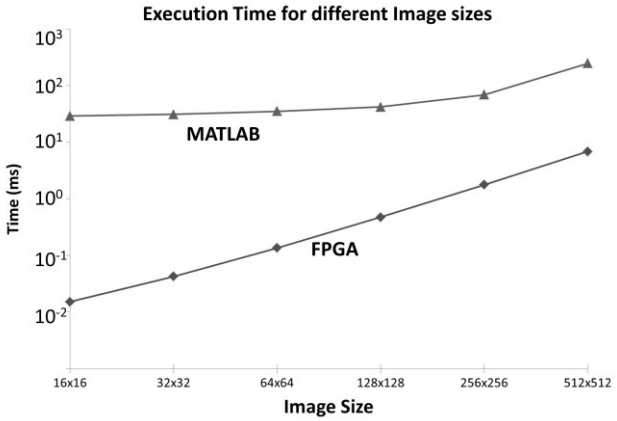

Figure 5 The execution time (ms) for different images sizes of the cup for both MATLAB and FPGA.

The data displayed in figure 5 shows the advantage of using the hardware implementation (FPGA) over using the software (MATLAB), especially for large image sizes. The code was implemented on a MATLAB platform on a desktop computer; with a 2.8-GHz Intel core quad processor, 4-Gb memory, the FPGA listed below running on 40MHz clock frequency, we notice a significant time increase when the size of the image goes over 256x256 in software, while it increases with same scale in hardware implementation.

Table 1 The FPGA used resources with maximum frequency and power.

Logic and Memory Rescores SPARTAN-6 xc6slx150t	Used	Available	Utilization
Number of Slice Registers	3,115	184,304	1%
Number of Slice LUTs	1,631	92,152	1%
Number of occupied Slices	422	23,038	1%
Maximum internal frequency	44.217MHz		
Est. Power Consumption	159 m W		

Table 1 shows the utilization of resources used in the design of the filter. It is shown that it utilized about 1% of the available resources on the FPGA device Spartan-6 XC6SLX150t-3, this due to the simple suggested operation used during the design of the filter.

Most visual prostheses will be wearable technology where higher resolutions are desirable. In our design architecture it is possible to add more filters working in parallel to each other, thus not increasing the latency, each filter consuming approximately 160mW.

IV. Conclusions

In this paper we have presented a processing platform to enhance the information flow in visual prosthesis. Although our focus is on optogenetic retinal approaches, the presented work is applicable to all forms of prosthesis where the visual return is significantly lower than normal. The aim of this pre-processing is to enhance and maximize the visual information included in the scene before stimulating the retina. Using our multispectral IR-visible approach can improve power and speed performance of the hardware by simplifying processing.

V. Acknowledgements

We would like to thank the European FP7 program for funding the OptoNeuro (249867) research program. Musa Al Yaman also like to thank the University of Jordan for research funding during his recently PhD program. Dr. Walid Al Atabany would also like to thank the Egyptian government for research funding during his recently completed PhD program. Finally we would like to thank Superflux for inspiring this work by jointly creating "the Song of the machine" documentary (http://www.superflux.in/work/song-machine)

VI. References

[1] V. Busskamp, et al., "Genetic Reactivation of Cone Photoreceptors Restores Visual Responses in Retinitis Pigmentosa," Science, vol. 329, pp. 413-417, Jul 23 2010.

[2] J. L. Stone*, et al.*, "Morphometric Analysis of Macular Photoreceptors and Ganglion-Cells in Retinas with Retinitis-Pigmentosa," *Archives of Ophthalmology,* vol. 110, pp. 1634-1639, Nov 1992.

[3] E. Zrenner, K. U. Bartz-Schmidt, H. Benav, D. Besch, A. Bruckmann, V. P. Gabel, et al., "Subretinal electronic chips allow blind patients to read letters and combine them to words," Proceedings of the Royal Society B-Biological Sciences, vol. 278, pp. 1489-1497, May 22 2011.

[4] P. Degenaar, N. Grossman, M. A. Memon, J. Burrone, M. Dawson, E. Drakakis, et al., "Optobionic vision--a new genetically enhanced light on retinal prosthesis," Journal of Neural Engineering, vol. 6, p. 035007, Jun 2009.

[5] D. J. B. Jan Müller, and Andreas Hierlemann, "Sub-millisecond closed-loop feedback stimulation between arbitrary sets of individual neurons," Front Neural Circuits, vol. 6, 2012.

[6] N. Grossman, K. Nikolic, C. Toumazou, and P. Degenaar, "Modeling Study of the Light Stimulation of a Neuron Cell With Channelrhodopsin-2 Mutants," IEEE Transactions on Biomedical Engineering, vol. 58, p. 9, 2011.

[7] P. Degenaar, B. McGovern, R. Berlinguer-Palmini, N. Vysokov, N. Grossman, V. Pohrer, et al., "Individually addressable optoelectronic arrays for optogenetic neural stimulation," in Biomedical Circuits and Systems Conference (BioCAS), 2010 IEEE, 2010, pp. 170-173.

[8] Al-Atabany WI, Memon MA, Downes SM, Degenaar PA. Designing and testing scene enhancement algorithms for patients with retina degenerative disorders. BioMedical Engineering Online 2010, 9, 27.

[9] W. Al-Atabany and P. Degenaar, "Scene optimization for optogenetic retinal prosthesis," in Biomedical Circuits and Systems Conference (BioCAS), 2011 IEEE.

[10] Al-Atabany, W.; Degenaar, P., "Efficient scene preparation and downscaling prior to stimulation in retinal prosthesis," Biomedical Circuits and Systems Conference (BioCAS), 2013 IEEE, pp.182-185.

[11] P. Perona and J. Malik, "Scale-Space and Edge-Detection Using Anisotropic Diffusion," IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 12, pp. 629-639, Jul 1990.