

HDL Implementation & Performance Comparison of Various Filtering Techniques Using FPGA

Ch. Sravana Vaidehi^{*1}, B. Murali Krishna^{#2}, G.L.Madhumati^{#3}, N. Kalyan^{#4},

*^{*1}Student of VLSI Systems Research Group, Department of Electronics and Communication Engineering, K.L University, Guntur AP – India
sravanavaidehi@gmail.com*

*^{#2}Asst. Professor VLSI Systems Research Group, Department of Electronics and Communication Engineering, KL University, AP – India
boppana.muralikrishna@gmail.com*

^{#3}Professor & H.O.D VLSI Systems Research Group, Department of Electronics and Communication Engineering, Dhanekula Institute of Engineering & Technology, AP – India

*^{#4}Student of VLSI Systems Research Group, Department of Electronics and Communication Engineering, K L University, AP – INDIA
kalyannallamothu4@gmail.com*

Abstract

In this paper we propose the application of various Filtering techniques to enhance the image using Verilog HDL and comparisons for various techniques in Xilinx and Cadence. This specifies the quality of an output by adapting new filter design techniques. The incoming signal in form of image pixel will be convoluted by various filtering techniques to define the edge and vale regions of image. The main characteristic of this paper was to read the input image and apply the generated filter coefficients obtained from the Matlab to various filtering techniques like Gaussian, Blur, and Sharpen., using HDL algorithm. Simulation results of HDL implementation of filtered coefficients were observed in Modelsim. The output image of HDL algorithm is obtained in Matlab. Physical View of implementation results are shown in VGA Display using Spartan3E FPGA Board. Area, Power and delay of various Filtering Techniques were analyzed and compared using Cadence RTL Compiler.

Keywords- Matlab, Filtering Techniques, Filtering Techniques, FPGA, VGA, Cadence RTL Compiler

INTRODUCTION

Field-Programmable Gate Arrays (FPGA's) are flexible and reusable high density circuits that can be easily re-configured by the designer, enabling the VLSI design, validation, and simulation cycle to be performed more quickly and less expensive [1]. The emerging market for image and video processing systems requires high-performance digital signal processing as well as low device costs appropriate for a volume applications. The main goal of this paper was to implement filtering techniques for different filters in Verilog Hardware Description Language. Matlab software for reading the input images, Model Sim simulator for simulating waveform and for applying filtering techniques, Xilinx Integrated Software Environment for VGA module is connected to a real time FPGA based hardware implementation. The goal of this project was to implement filtering techniques for different filters. Filters like Gaussian, Blur, and Sharpening are used in this paper. In recent years, sharpening filtering technique is used as image enhancement. Gaussian filter is used as canny edge detector, GFSK. The paper is organized as follows: Section 2 presents an experimental survey of various filters. Section 3 presents a design flow. The design methodology was discussed in Section 4. Section 5 describes the implementation of hardware design on FPGA. The experimental results are given in Section 6. Section 7, 8 makes the concluding remarks and references.

EXPERIMENTAL SURVEY

The process that cleans up appearances and allows for selective highlighting of specific information is called Filtering. A number of techniques are available and the best options can depend on the image and how it will be used. In this paper following filtering techniques are used:

1. Gaussian filter
2. Blur filter
3. Sharpening filter

A Gaussian filter [2] is a Gaussian function as its impulse response (or an approximation to it). The properties of Gaussian filters having step function as a input with no overshoot and minimizing the rise and fall time. This behaviour is nearly equals to the connected fact that the minimum possible group delay for the Gaussian filter. The ideal time domain filter, such as the sinc is as oscilloscopes and the digital telecommunication systems.

Mathematically a Gaussian filter converts the input signal by convolution with a Gaussian function.

In one dimension, the Gaussian function is

$$G(x) = \frac{1}{\sqrt{2\pi(\sigma^2)}} \cdot e^{-\frac{x^2}{2\sigma^2}}$$

Where σ is the standard deviation of the distribution.

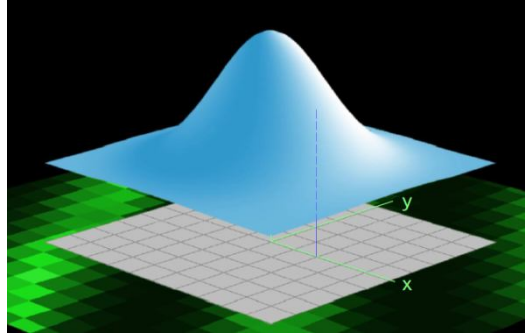


Fig 1: Gaussian filter

In two dimensions, Gaussian function is given as

$$G(x, y) = \frac{1}{2\pi\sigma^2} \cdot e^{-\frac{x^2+y^2}{\sigma^2}}$$

The 3x3 mask of the Gaussian filter is given as

$$\begin{bmatrix} 0.0113 & 0.0838 & 0.0113 \\ 0.0838 & 0.6198 & 0.0838 \\ 0.0113 & 0.0838 & 0.0113 \end{bmatrix}$$

A blur filter is also called as low pass filter [3] or smoothing filter. A filter that average intensity with rapid change is known as blur filter. The blur filter summarizes the average of a pixel and all of its eight immediate neighbors. The result is replaced by the original value of the pixel. For every pixel in the image this process is repeated. Why we need blur image? Even the camera is good; images can be noisy and always adds an amount of snow into the image. Light of its statistical nature also contributes noise into the image. Each pixel generates its own independent noise so that values of each pixel change from pixel to pixel. The image from the telescope isn't unrelated in this fashion because real images are spread over many pixels. So the blur filter affects the noise more than it does the image. Gradual changes can be seen by suppressing the noise which are invisible before. Therefore a blur filter can sometimes be used to bring out faint details that were smothered by noise.

By drawing a “convolution kernel” filtering can be visualized. A small grid showing how a pixel's filtered value depends on its neighbour's pixel values refer to the kernel. To perform a blur filter by simply averaging adjacent pixel's.

The following kernel is used:

$$\begin{bmatrix} 1/9 & 1/9 & 1/9 \\ 1/9 & 1/9 & 1/9 \\ 1/9 & 1/9 & 1/9 \end{bmatrix}$$

Human insight is extremely responsive to edges and fine details of an image, due to they are composed primarily by high frequency components, if the high frequency components are attenuated, the visual quality of an image can be largely degraded. In

contrast, enhancing the high frequency components of an image leads to an improvement in the visual quality. The enhancement technique that highlights edges and fine details in edges is known as Image sharpening [4]. Image sharpening is widely use in printing and photographic industries for increasing the local contrast and sharpening the images.

Kernel for sharpening filter is given as:

$$\begin{bmatrix} -0.1667 & -0.6667 & -0.1667 \\ -0.6667 & 4.3333 & -0.6667 \\ -0.1667 & -0.6667 & -0.1667 \end{bmatrix}$$

DESIGN FLOW FOR IMAGE PROCESSING

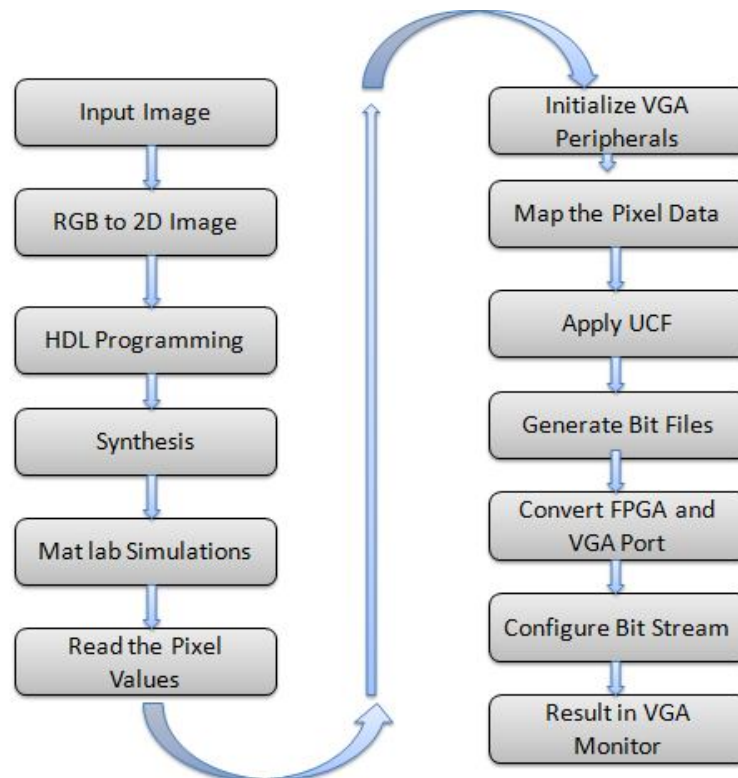


Fig 2: design flow for image processing

The input image is taken from Matlab work space. If the image is a colour image than it is converted into gray scale. If the image is 3D then it is converted into 2D image this is done in one stage. Then the algorithm is run in the Matlab [5] and Modelsim [6] one after other after getting simulating results in the modelsim, the filtered output is shown in the Matlab. When filtered image is generated, the pixel values of the input image are stored in the text file named as the file out of FPGA folder. Than the pixel values were read from the text file. Then initialize VGA

peripherals, map the pixel data and apply the user constrain files and apply the bit stream. Now connect the FPGA and VGA port and configure the bit stream now results can be shown on the VGA display.

DESIGN METHODOLOGY

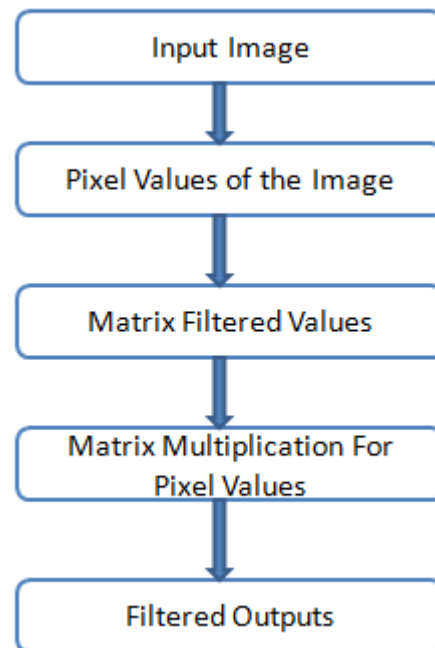


Fig 3: Design Methodology of matlab image

An algorithm was developed in Matlab to process an image for filtering operations. The algorithm describes the pixel values of the selected image and stores them in out file. Matrix values are different for varies filters. Matrix multiplication is done for pixel values of the image and the matrix values of the filter. The output of the matrix multiplication will be the filtered output.

IMPLEMENTATION

Implemented paper is in Xilinx Spartan3E FPGA Board. A Verilog HDL program has been written in modelsim and Xilinx ISE for the above filtering techniques and VGA display. A Matlab coding is used for reading the input image. Synthesis and Implementation involves net list, bit stream generation, and download the design into FPGA through Xilinx IMPACT. The output of the different filtered image was displayed on the VGA monitor.

RESULTS

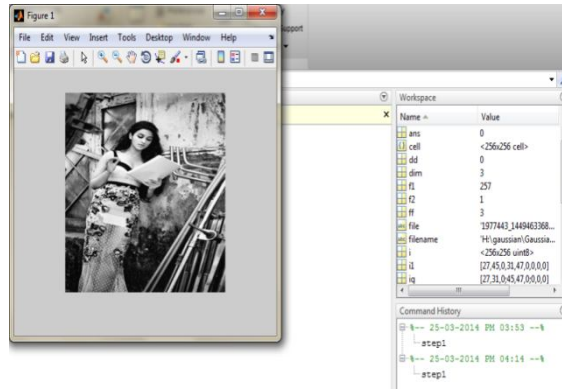


Fig 4: Blurred Input Image



Fig 5: FPGA Output For Blurred Filter

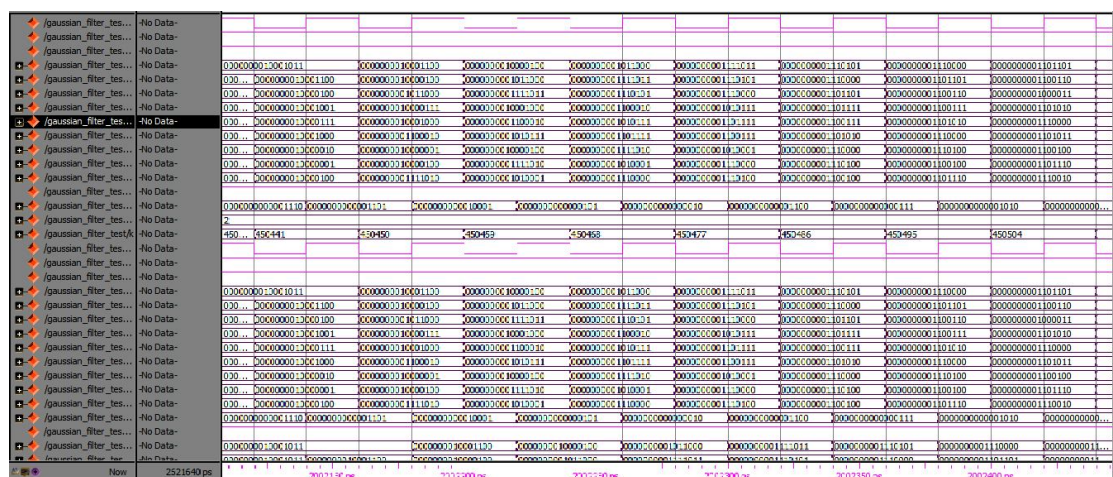


Fig 6: Simulation Result for Blurred Filter

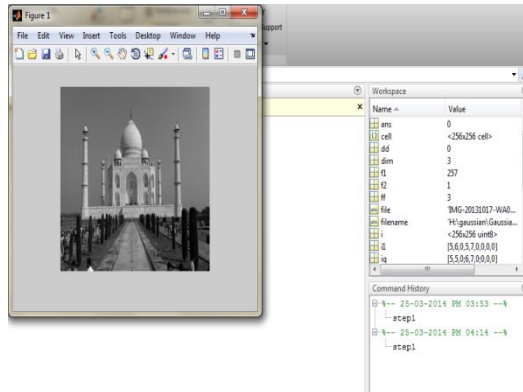


Fig 7: Gaussian Input Image

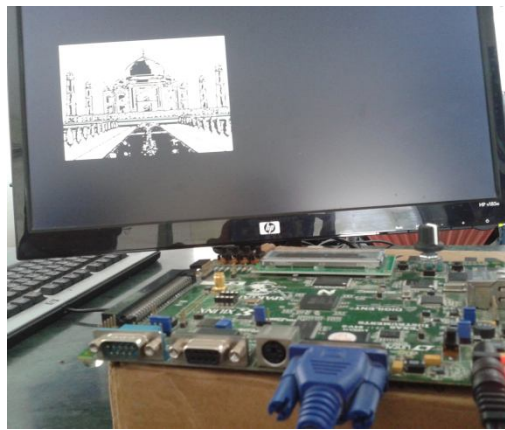


Fig 8: FPGA Output For Gaussian Filter

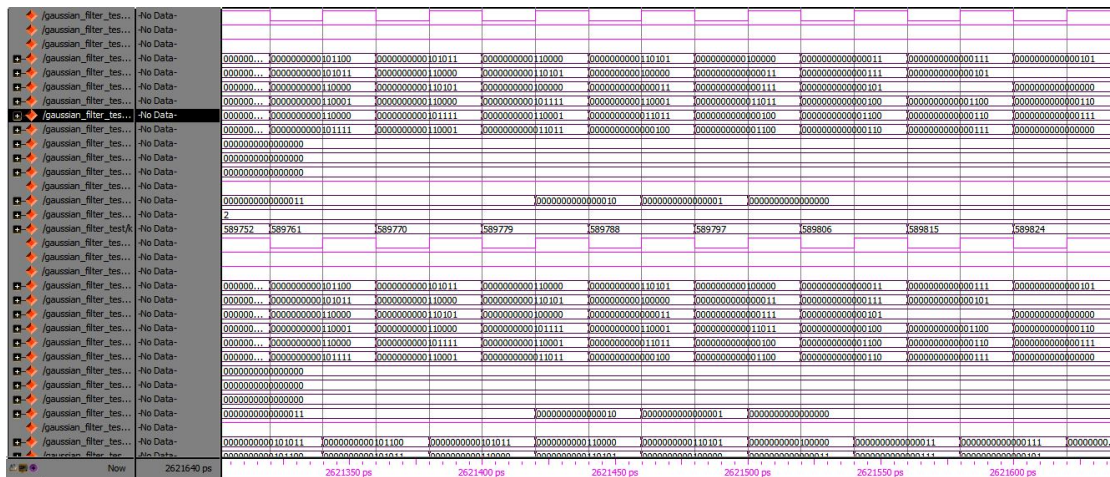


Fig 9: Simulation Result For Gaussian Filter

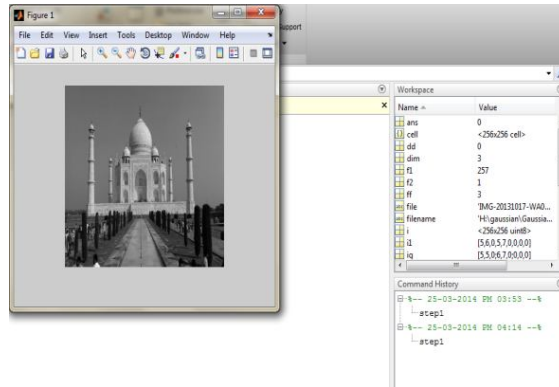


Fig 10: Sharpening Filter Input Image

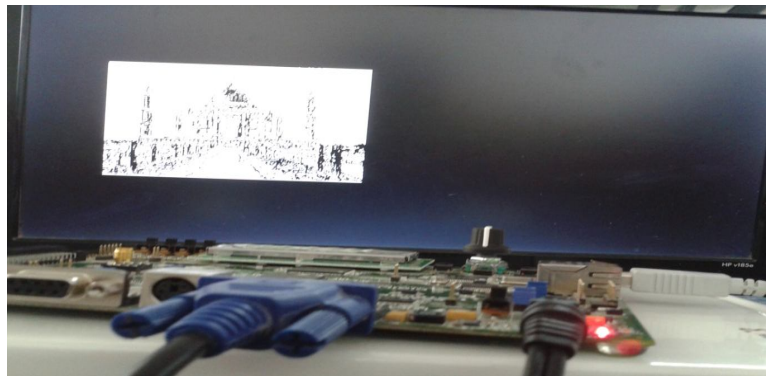


Fig 11: FPGA Output For Sharpening Filter

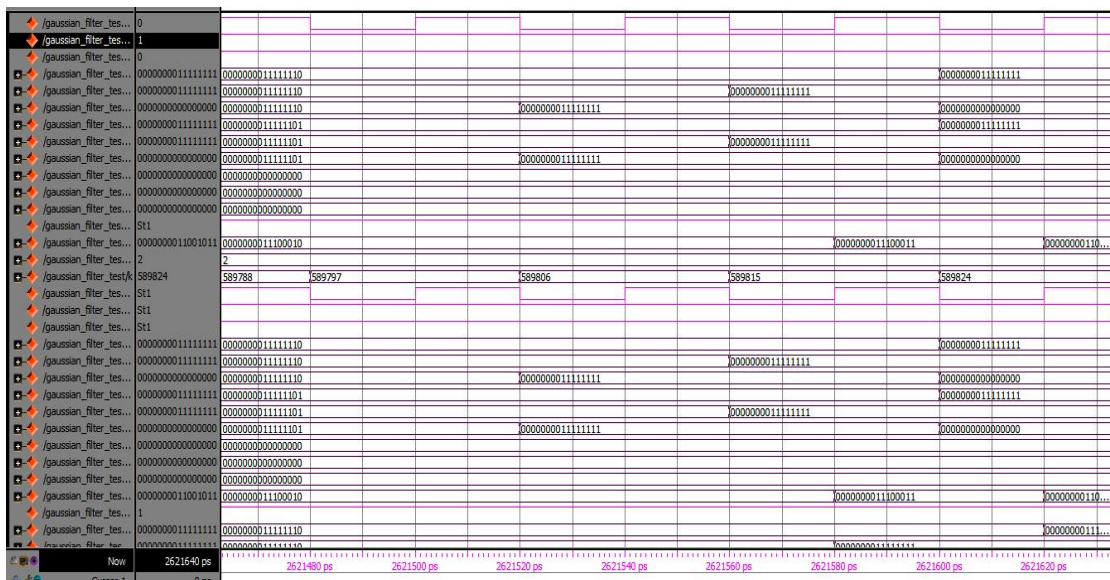


Figure12: Simulation Result for Sharpening Filter

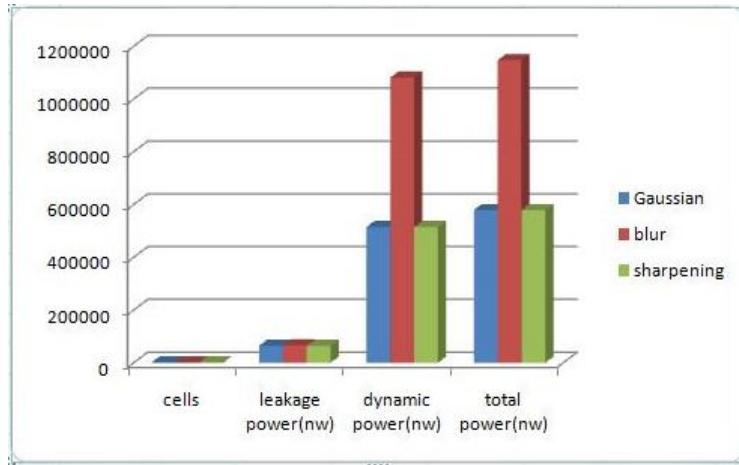


Figure 13: Comparisons for various Techniques for power

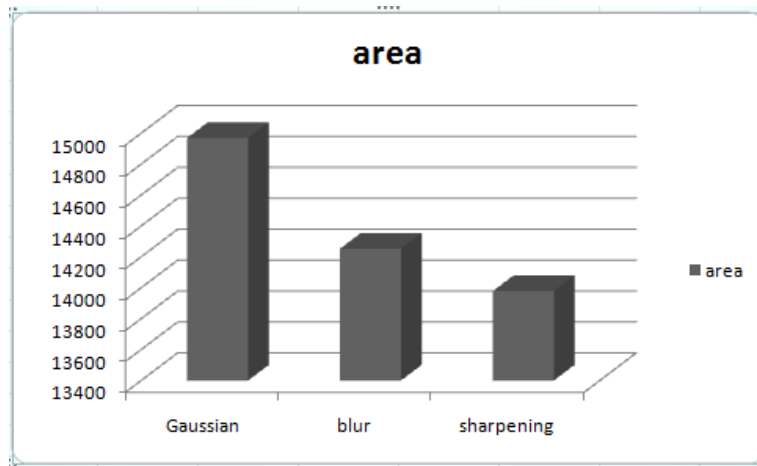


Figure 14: Comparisons for various Techniques for Area

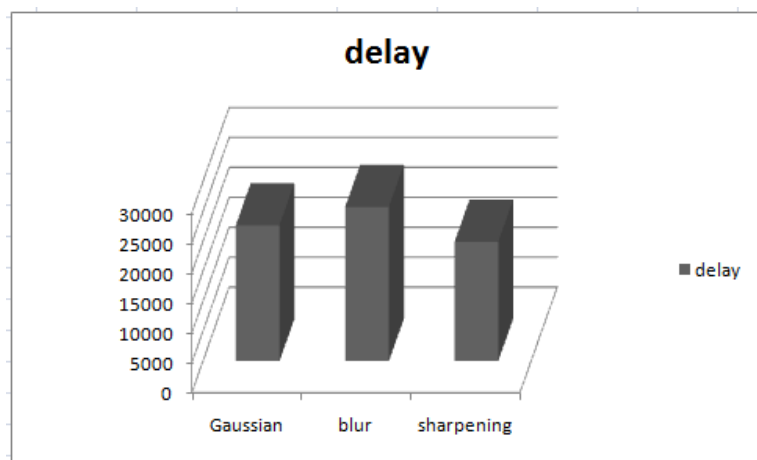


Figure 15: Comparison for Delay

CONCLUSION

The purpose of this paper was to illustrate the different type of filters in image processing and comparison of various techniques in Cadence. This design is implemented in the device Spartan 3E FPGA. The advantage of this paper is by using the pixel values image was shown in real time using HDL programming. By the results obtain we can state verilog HDL program can be used efficiently in image processing as considerable for other programmable styles. By the results obtain in cadence we can state that the sharpening filter has less area, delay and Power.

REFERENCES

- [1] A. K. Jain, *Fundamentals of Digital Image Processing*. Englewood Cliffs, NJ: Prentice-Hall, 1989.
- [2] Fabrizio Russo, *Member, IEEE*, IEEE TRANSACTIONS ON INSTRUMENTATION AND MEASUREMENT, VOL. 52, NO. 4, AUGUST 2003
- [3] IEEE TRANSACTIONS ON IMAGE PROCESSING, VOL. 8, NO. 12, DECEMBER 1999
- [4] www.ijscce.org/attachments/File/v3i3/C1716073313.pdf
- [5] S. K. Mitra and G. Sicuranza, Eds., *Nonlinear Image Processing*. New York: Academic, 2000.
- [6] http://www.mentor.com/company/higher_ed/
- [7] I. Pitas and A. N.Venetsanopoulos, *Nonlinear Digital Filters: Principles and Applications*. Norwell, MA: Kluwer, 1990.
- [8] G. Ramponi, "Image processing with rational operators: Noise smoothing and anisotropic diffusion," *Proc. IEEE-EURASIP 2nd Int Symp. Image Signal Processing Anal.*, June 2001.
- [9] F. Russo, "An image enhancement technique combining sharpening and noise reduction," *IEEE Trans. Instrum. Meas.*, vol. 51, pp. 824–828, Aug. 2002.
- [10] J. S. Jin, Y. Wang, and J. Hiller, "An adaptive nonlinear diffusion algorithm for filtering medical images," *IEEE Trans. Inform. Technol. Biomed.*, vol. 4, pp. 298–305, Dec. 2000.
- [11] M. J. Black, G. Sapiro, D. H. Marimont, and D. Heeger, "Robust anisotropic diffusion," *IEEE Trans. Image Processing*, vol. 7, pp 421–432, Mar. 1998.
- [12] Palinitkar S (2003) Verilog HDL: a guide to digital design and synthesis, 2nd edn. Prentice Hall, Palo Alto
- [13] www.cyanogen.com/help/maximdl/Low-Pass_Filtering.htm
- [14] Muhammad Aljuaid and Halim Yanikomeroglu, *Member IEEE* IEEE TRANSACTIONS ON VEHICULAR TECHNOLOGY, VOL. 59, NO. 9, NOVEMBER
- [15] Xiang Zhu, *Member, IEEE*, Scott Cohen, *Member, IEEE*, Stephen Schiller, *Member, IEEE*, and Peyman Milanfar, *Fellow, IEEE* IEEE TRANSACTIONS ON IMAGE PROCESSING, VOL. 22, NO. 12, DECEMBER
- [16] Akira Iwasaki and Eimei Oyama IEEE TRANSACTIONS ON GEOSCIENCE AND REMOTE SENSING, VOL. 43, NO. 12, DECEMBER 2005