

## Simulation of Steerable Gaussian Smoothers using VHDL

*Sharanabasava, Syed Gilani Pasha*

M.Tech Scholar

Sri K S Raju Institute of technology & Science  
Moinabad, Hyderabad

Email: sharanugilke034@gmail.com

Assistant Processor

Dept. of Electronics & Communication Engineering  
Central University of Karnataka  
Gulbarba

**Abstract**— Smoothing filters have been widely used in image and video analysis. Also directional smoothers useful in motion analysis, edge detection, line parameter estimation, and texture analysis. Such particular applications require the use of several different angles oriented directional filters. For real time applications, hardware devices having capability of parallel processing can be used. The steerability is the property, in which several filtering operations outputs are linearly combined to achieve output of a directional filter which is arbitrarily oriented. Smoothing filters has a property of steerability, which implies that the outputs of several filtering operations can be linearly combined in order to produce the output of a directional filter at an arbitrary orientation. There are several efficient FPGA implementations of the convolution operation for non-separable and separable have been presented in the literature, research on steerable filter implementations on FPGA is limited. In this paper, we present implementations of steerable Gaussian smoothers using VHDL and simulation is carried out using Model SIM software.

**Keywords:** - Steerability; Directional filter; FPGAs; Gaussian Filters; Separable convolution;

[7], [8 ]. All the hardware architectural models are prototyped, a device technology of Xilinx FPGA platform

### I. INTRODUCTION

Directional or orientation filters [1] are widely used for motion analysis, edge detection and texture analysis[9]. The image components, such as edges and lines, can be characterized by a set of parameters including position, orientation, width or size. One method for obtaining the response of a filter at any arbitrary position and orientation is to tune the filter to all possible positions and orientations. However, such an approach requires a large number of computations, and is thus not easily implementable in real time. An alternative, more efficient way, is to design a family of basis filters [2] [3], so that a filter tuned at an arbitrary position or orientation can be represented as a linear combination of these basis filters. Therefore, the output of the steerable filter can be expressed as a weighted sum of the basis filter outputs. Directional filters having these properties are called "Steerable Filters". This project work emphasis is on the implementation of proposed computationally efficient separable and steerable Gaussian smoothers on a FPGA platform.

The aim of this project is to design and implement a Steerable (or directional) Gaussian Smoothers using VHDL. The main objective of this project is to develop an efficient architecture for Steerable Gaussian smoothers simulated in VHDL.

The main contribution in this work is the design of directional Gaussian smoothers [2] using VHDL and implemented on FPGA. A VHDL model is developed for a test image of  $7 \times 7$  and a Gaussian mask of  $3 \times 3$ . Based on the simulation results and logic utilization, we implemented the convolution operation similar to the techniques presented in

### II. LITERATURE SURVEY

In [8], Area-Efficient 2-D shift-variant convolvers for FPGA-based digital image processing are proposed. They proposed several novel FPGA-efficient architectures for generating a moving window over a row-wise print path, and provided a criteria to choose the optimum one for any design point. Hui Zhang et. al. proposed a Multiwindow Partial Buffering Scheme for FPGA Based 2-D Convolver [7]. Erke Shang et. al. in [9] puts forth about Architectures for Generalized 2D FIR Filtering using Separable Filter Structures. The problem of generalized 2D FIR filtering for large filter kernel sizes can be computationally prohibitive when required in real-time. In [4], C.S Bouganis et. al. emphasized on steerable. pyramid wavelet construction for image decomposition and feature detection, and its implementation on FPGA. Erke Shang et.al. used steerable filters for lane detection and implemented it on FPGA[5]. If the approach presented in [4] and [5] are used then large number of basis filters would be required. Instead, regardless of the desired angular resolution, Gaussian smoothers can be steered via the application of three 1D filtering operations [16].

### III. PROPOSED WORK

In this paper, we propose to simulate steerable filtering techniques using VHDL and it is implemented on FPGA efficiently. The design is divided into steps, first the Image smoothing is done by convolving the original image with a Gaussian mask and at second stage Optimizing and pipelining is done which improves the implementation efficiency. In this work, both separable simulation and implementations, unpipelined and pipelined were used. The Gaussian mask can be rotated in different directions to obtain steerable output at different directions. The two different

separable techniques for the purpose of implementing the filter mask are unpipelined and a pipelined. The block diagram representation of unpipelined separable convolution method is shown in Fig. 1.

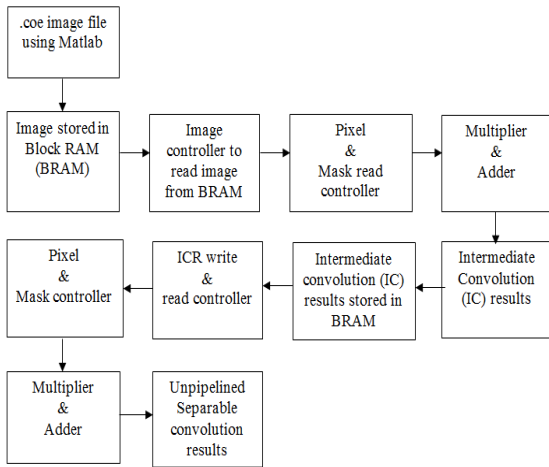


Fig.1. Block diagram representation of unpipelined separable convolution

In this method, the image is first convolved with the vertical 1D Gaussian mask and second with the horizontal Gaussian mask. At each adder stage rescaling is performed to ensure that the intensity value of the output pixels does not exceed 255 (8 bits). Using this method, 2 clock cycles are required to obtain the required output pixel, independent of image and mask sizes.

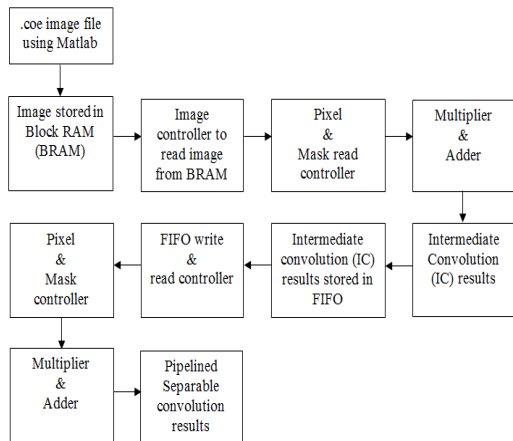


Fig.2. Block diagram representation of pipelined separable Convolution

A pipelining technique can be used to obtain higher throughput. Partial intermediate convolution results i.e.  $N$  rows and  $P$  columns of the image are stored in a FIFO or 2D array. This method requires only 1 clock cycle per pixel, regardless of the image and filter sizes. Fig.2 shows the block diagram representation of this method. The two separable techniques were implemented on Xilinx Spartan 2 Pro [12] for an input image of  $158 \times 158$  and a Gaussian mask of  $7 \times 7$ .

The block diagram representation of the steerable implementation is shown in Fig. 3. The images smoothing is done using unpipepined and pipepined separable methods and smoothen images are stored in BRAMs. The controller read pixels in horizontal and vertical directions from the smoothed image. A pixel and mask controller gives the pixels to the multiplier and adder blocks. At the adder stage a rescaling

step is performed to ensure that the intensity value of the output pixels remains between 0 and 255. First, a small test image of  $7 \times 7$  and a Gaussian mask of  $3 \times 3$  were chosen for performing the convolution operation. The two dimensional convolution operation was implemented using three different approaches which are listed below:-

- 1) General two dimensional convolution Method
- 2) Separable convolution method 1 (using multiple BRAMs)
- 3) Separable convolution method 2 (using FIFO)

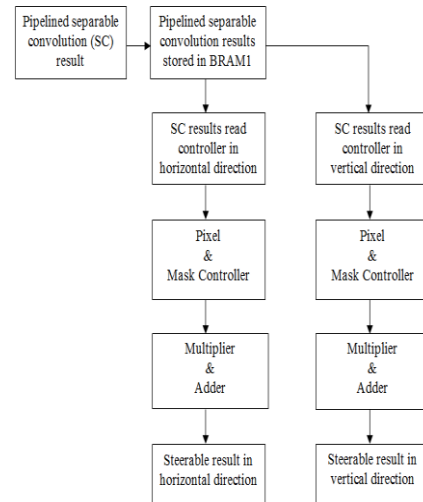


Fig.3. Block diagram representation of steerable implementation in horizontal and vertical directions.

For all methods explained below, a test  $7 \times 7$  image and a  $3 \times 3$  Gaussian mas with mean = 0 and standard deviation = 1 and normalizing factor  $N = 0.0016$  are considered. Each test image pixel is represented using 16 bits and each mask value is also represented using 16 bits. A  $7 \times 7$  test image and a  $3 \times 3$  Gaussian mask are shown below:-

#### IV. SIMULATION RESULTS & DISCUSSIONS

In this section we describe details of simulation results are obtained for an image of size  $158 \times 158$ , and a separable Gaussian mask of  $7 \times 7$  using the steerable filter implementation using Xilinx tools. The Fig.4 shows the input image taken for simulation. In this simulation two separable methods are used. The isotropic filtering, which is equivalent to two 1D filtering operations (a horizontal and a vertical), and a directional 1D filtering operation represents an efficient steerable filtering implementation. In this project work, results are obtained for directional Gaussian smoothers with standard deviation  $\sigma_x = 3$  and  $\sigma_y = 5$ . These two standard deviations represent the filter extent with respect to the direction of filter ( $\sigma_y$ ) and the angle vertical to it ( $\sigma_x$ ), which can be arbitrary. On the bases of two standard deviation values, the directional 1D Gaussian mask is of size  $1 \times 9$ , and the 1D masks associated to the isotropic Gaussian mask are of size  $1 \times 7$  and  $7 \times 1$ . The design is implemented and simulated on Xilinx Spartan 2 FPGA for maximum clock frequency of 100MHz, in VHDL language using Xilinx ISE 14.7 Software. Fig.5, Fig.6 Fig.7 and Fig.8 shows the simulations outputs.



Fig.4 shows input image

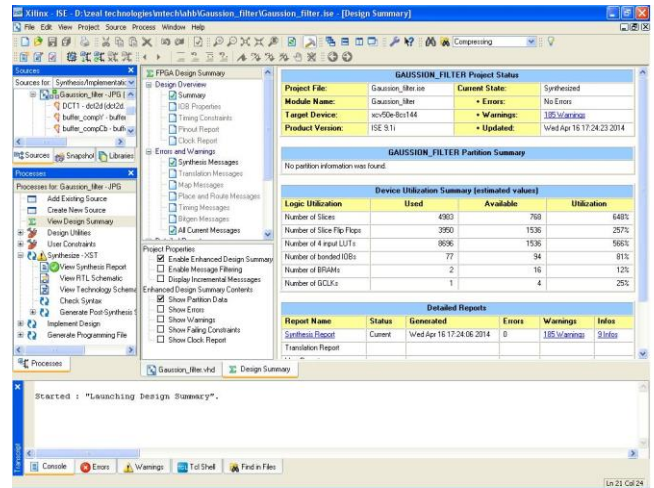


Fig.6 shows Gaussian synthesis report

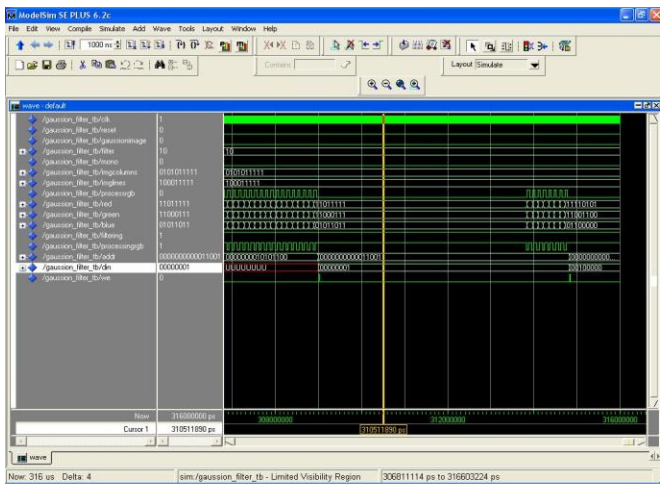


Fig.5 shows Gaussian filter output

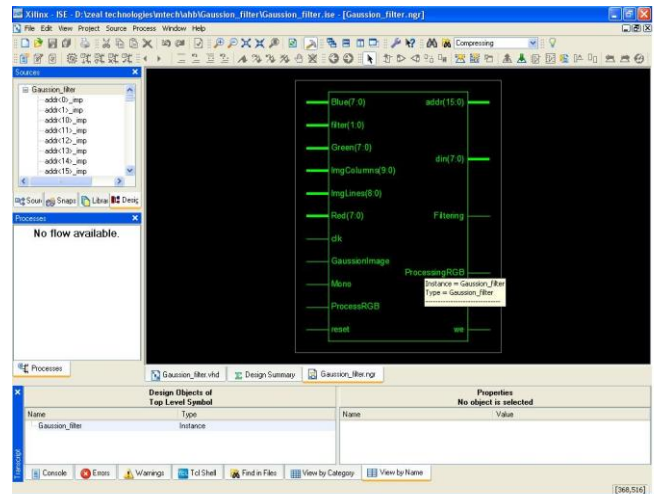


Fig.7 shows Gaussian filter port diagram  
The RTL schematic that was obtained using ISE10.1 is shown in Fig. 5 for two steerable filter directions.

Steerable implementation in horizontal and vertical directions using the unpipelined separable convolution method requires fewer resources and a throughput of 3 clocks cycles per pixel is achieved.

- Steerable implementation in horizontal and vertical direction using the pipelined separable convolution method requires huge resources and increase with the size of the input image. However an optimum performance of 2 clocks cycles per pixel is achieved. Through this comparison, it can be observed that there is a trade-off among resource utilization and processing time. Hence the architecture selection can be done depending on the user constraints on whether to go with less resources and more processing time or more resources and less processing time.

It can be seen that the pipelined steerable implementation utilizes almost all the resources available on the board. This can be avoided by storing the original and intermediate processed images in the SDRAM or Flash Memory, which is part of our future work

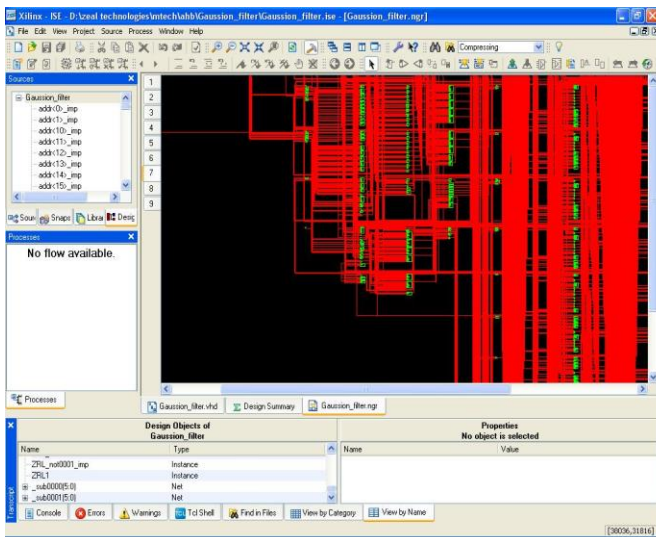


Fig.8 shows Gaussian RTL schematic

## V. CONCLUSION AND FUTURE WORK

In this paper, an efficient steerable Gaussian filter implementation using VHDL on FPGA has been design. Similarly to previous techniques, the proposed approach takes advantage of the separable nature of isotropic Gaussian filters. Simulation results confirm that the pipelined steerable filter implementation on FPGA (100MHz clock) is significantly faster compared to a C implementation [1] executed on a PC with significantly higher clock speed (Dual Core 2 2.33-GHz).

In future work, the technique will be modified to reduce the number of clock cycles per pixel down to 1, by pipelining the pass of the third 1D filter.

## REFERENCES

- [1] Dimitrios Charalampidis, "Efficient Directional Gaussian Smoothers", IEEE Geoscience and Remote Sensing Letters, July 2009.
- [2] V. Lakshmanan, "A Separable Filter for Directional Smoothing", IEEE Geoscience and Remote Sensing Letters, July 2004.
- [3] W.Freeman and E.Adelson, "The Design and Use of Steerable Filters", IEEE Transaction on Pattern Analysis and Machine Intelligence, pp. 891-906, 1991.
- [4] C.S Bouganis, P.Y.K Cheung, J.Ng and A. Bharath, "A Steerable Complex Wavelet Construction and its Implementation on FPGA", in Proc. International Conference on Field Programmable Logic and Applications, pp.394-403, 2004
- [5] Erke Shang, Jian Li, Xiangjing An and Hangen He, "Lane Detection using Steerable Filters and FPGA Based Implementation", International Conference on Image and Graphics, 2011.
- [6] S. Kestur, D. Dantara and V. Narayanan, "A streaming FPGA Implementation of a Steerable Filter for Real Time Applications(abstract only)", Proceeding of 19<sup>th</sup> ACM/SIGDA International Symposium on FPGA, 2011.

[7] Hui Zhang, Mingxin Xia, and Guangshu Hu, "A Multiwindow Partial Buffering Scheme for FPGA Based 2-D Convolvers, IEEE Transaction on Circuits and Systems, Feb07.

[8] Francisco Cardells-Tormo, Pep-Lluis Molinet, "Area Efficient 2-D Shift Variant Convolvers for FPGA Based Digital Image Processing", IEEE Transaction on Circuits and Systems, Feb 2006.

[9] D. Venkateshwar Rao and M. Venkatesan, "Implementation and Evaluation of Image Processing Algorithms on Reconfigurable Architecture using C-based Hardware Descriptive Languages", International Journal of Engineering and Applied Sciences, 2006.

[10] M.S. Andrews, "Architectures for Generalized 2D FIR Filtering using Separable Filter Structures", Proceeding of Acoustics, Speech and Signal Processing, 1999.

[11].A. Joginipelly, Alvaro Varela, Dimitrios Charalampidis, et.al. "Efficient FPGA Implementation of Steerable Gaussian Smoothers", 44th IEEE Southeastern Symposium on System Theory University of North Florida, Jacksonville, FL,2012, .78-82.

[12] Xilinx,"Virtex-II Pro Platform FPGA User Guide-UG012",v4.2, Nov 2007

[13] William T. Freeman and Edward H. Adelson, "The Design and Use of Steerable Filters", IEEE Transaction of Pattern Analysis and Machine Intelligence, Sept 1991.

[14]. C. Chou, S. Mohana krishnan, J. Evans, "FPGA Implementation of Digital Filters", Proc. ICSPAT, 1993

[15]. Dr. Haled Benkrid, Mr. Samir Belkacemi, "Design and Implementation of a 2D Convolution Core for Video Applications on FPGAs", International Workshop on Digital and Computational Video, Nov 2002.