International Journal of Multidisciplinary Research and Development

International Journal of Multidisciplinary Research and Development

Volume: 2, Issue: 6, 360-365
June 2015
www.allsubjectjournal.com
e-ISSN: 2349-4182
p-ISSN: 2349-5979
Impact Factor: 3.762

Madan Veer
Department of Electronics & Telecommunication, RMDSSOE, Savitribai Phule Pune University, Warje, Pune, 411058, India

R.U. Shekokar
Department of Electronics & Telecommunication, RMDSSOE, Savitribai Phule Pune University, Warje, Pune, 411058, India

Correspondence:
Madan Veer
Department of Electronics & Telecommunication, RMDSSOE, Savitribai Phule Pune University, Warje, Pune, 411058, India

FPGA in Loop Implementation using Sobel Algorithm on Detected Object

Madan Veer, R. U. Shekokar

Abstract
This paper gives novel implementation of Object detection and edge detection technique using FPGA. Speeded-Up Robust Features (SURF) algorithm is used to find blob features. SURF approximates or even outperforms previously proposed schemes with respect to repeatability, distinctiveness, and robustness, yet can be computed and compared much faster. The key points inside each super pixel are estimated using the Speed-Up Robust Feature (SURF). The MSAC algorithm is a variant of the Random Sample Consensus (RANSAC) algorithm. These key points are then used to carry out the matching task for every detected key points of a scene in-side the estimated super pixels. This project presents an algorithm for detecting a specific object based on finding point correspondences between the reference and the target image. It can detect objects despite a scale change or in-plane rotation. It is also robust to small amount of out-of-plane rotation and occlusion. This method of object detection works best for objects that exhibit non-repeating texture patterns, which give rise to unique feature matches. This technique is not likely to work well for uniformly-colored objects, or for objects containing repeating patterns. Note that this algorithm is designed for detecting a specific object, for example, the elephant in the reference image, rather than any elephant. Also I have done FPGA in loop implementation on Soble Edge Detection algorithm analyzed same algorithm on Zynq Zybo, Virtex5, Virtex 4.

Keywords: SURF, MSAC, SOBEL ALGORITHM, SOBEL FILTER, FPGA

1. Introduction

1.1 Object detection

Feature detection is the process where we automatically examine an image to extract features that are unique to the objects in the image, in such a manner that we are able to detect an object based on its features in different images. This detection should ideally be possible when the image shows the object with different transformations, mainly scale and rotation, or when parts of the object are occluded. The processes can be divided in to 3 overall steps.

Detection: Automatically identify interesting features, interest points this must be done robustly. The same feature should always be detected regardless of viewpoint.

Description: Each interest point should have a unique description that does not depend on the features scale and rotation.

Matching: Given and input image, determine which objects it contains, and possibly a transformation of the object, based on predetermined interest points.

This paper presents an algorithm for detecting a specific object based on finding point correspondences between the reference and the target image. It can detect objects despite a scale change or in-plane rotation. It is also robust to small amount of out-of-plane rotation and occlusion. This method of object detection works best for objects that exhibit non-repeating texture patterns, which give rise to unique feature matches. This technique is not likely to work well for uniformly-colored objects, or for objects containing repeating patterns. Note that this algorithm is designed for detecting a specific object, for example, the elephant in the reference image, rather than any elephant.

1.2 Sobel edge detection

Sobel edge detection is a classical algorithm in the field of image and video processing for the extraction of object edges. Edge detection using Sobel operators works on the premise of computing an estimate of the first derivative of an image to extract edge information (Gonzalez, 2001). By computing the x- and y-direction derivatives of a specific pixel against a

neighborhood of surrounding pixels, it is possible to extract the boundary between two distinct elements in an image. Because of the computational load of calculating derivatives using squaring and square root operators, Sobel edge detection algorithm is considered as a computational bottleneck.
This paper provides a methodology for migrating Sobel edge detection algorithm from a processor onto the FPGA logic or on the GPU.

1.3 FPGA in Loop Implementation

FPGA-in-the-loop (FIL) simulation provides the capability to use Simulink or MATLAB software for testing designs in real hardware for any existing HDL code. The HDL code can be either manually written or software generated from a model subsystem. Similarly, FPGA can achieve extremely high performance in many applications in spite of its low operational frequency. In this context, we will define a methodology to implement automatically and optimize algorithms for image processing in complex electronic systems. Thus, we choose the edge detection since the data of edge detection is very large so the speed of image processing is a difficult problem. FIL performs the following processes when it creates the block or System object:

- Generates a FIL block or FIL System object that represents the HDL code Provides synthesis, logical mapping, place-and-route (PAR),
- Programming file generation, and communications channel.
- Loads the design onto an FPGA All these capabilities are specifically designed for a particular board and tailored to your RTL code.
- As part of FIL simulation, the block or System object and your model or application: Transmits data from Simulink or MATLAB to the FPGA Receives data from the FPGA

2. Object Detection

2.1

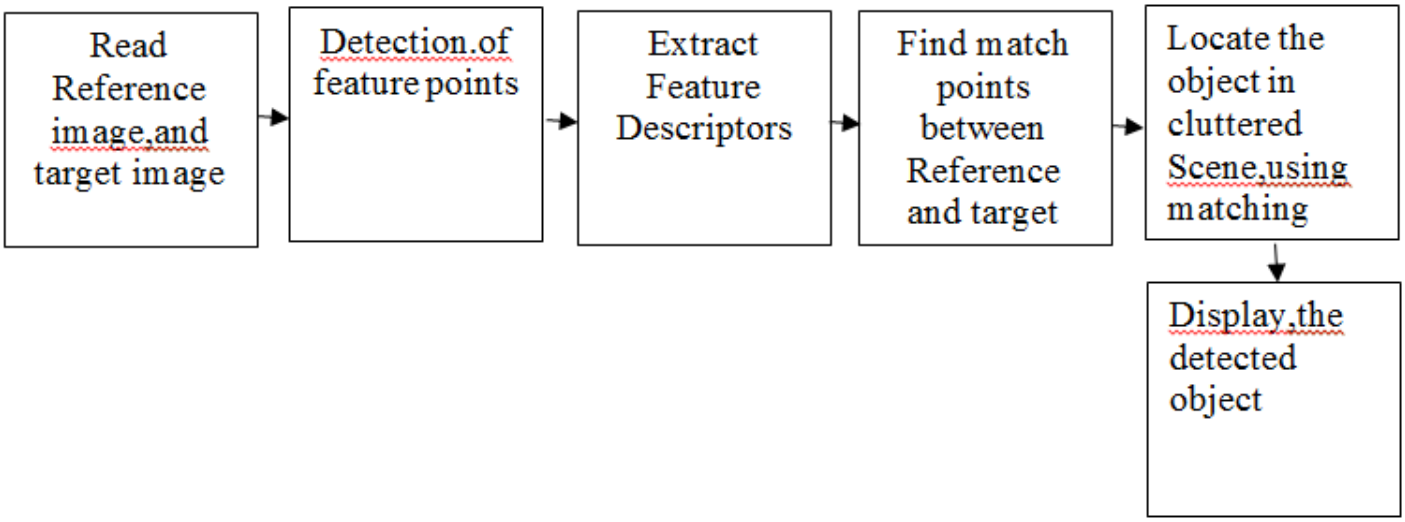


Fig 1: Algorithm for Detection of objects in cluttered Scene

2.2 Steps

Step 1: Read Images
-Read the target image containing a cluttered scene.
Step 2: Detect Feature Points
- Detect feature points in both images.
-Visualize the strongest feature points found in the reference image.
-100 Strongest Feature Points from box image.
- Visualize the strongest feature points found in the target image.
- 300 Strongest Feature Points from box image.
Step 3: Extract feature descriptors at the interest points in both images.
Step 4: Match the features using their descriptors & Display.
Step 5: Locate the Object in the Scene Using Putative Matches.
- Display the matching point pairs with the outliers removed.
- Get the bounding polygon of the reference image.
- Transform the polygon into the coordinate system of the target image.
- The transformed polygon indicates the location of the object in the scene.
Step 6: Display the detected object.
Step 7: Detect another Object
- Detect a second object by using the same steps as before.
- Read an image containing the second object of interest.
- Detect and visualize feature points.
Step 8: Extract feature descriptors.
Step 9: Match Features.
- Display putatively matched features.
Step 10: Estimate Geometric Transformation and Eliminate Outliers.
Step 11: Display Both Objects
-Detected Elephant and Box.

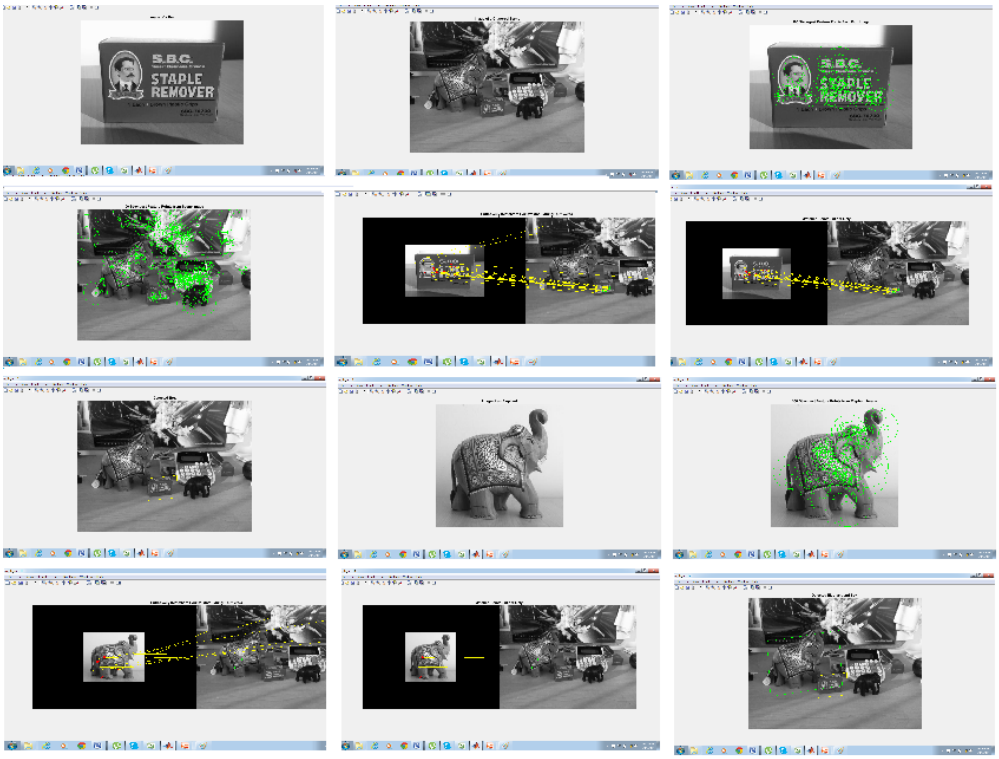


Fig 2: Object Detection in Cluttered Scene

3. Sobel Edge Detection

3.1 SOBEL Introduction

In Sobel Edge Detection there are two masks, one mask identifies the horizontal edges and the other mask identifies the vertical edges The mask which finds the horizontal edges that is equivalent to having the gradient in vertical direction and the mask which computes the vertical edges is equivalent to taking in the gradient in horizontal direction.

3.2 SOBEL Implementation Steps

- The input image is first converted to gray scaled image.
- Traverse through entire image.
- For each pixel in the image we will take a window of 3*3 pixel and multiply it the given template for matrix.
- Then we will calculate the G using formula. $\sqrt{Gx^2 + Gy^2}$

-1	0	+1	+1	+2	+1
-2	0	+2	0	0	0
-1	0	+1	-1	-2	-1

X Y

Apply the templates to a 3x3 filter window.

a1	a2	a3
a4	a5	a6
a7	a8	a9

Fig 3: 3x3 Filter Window

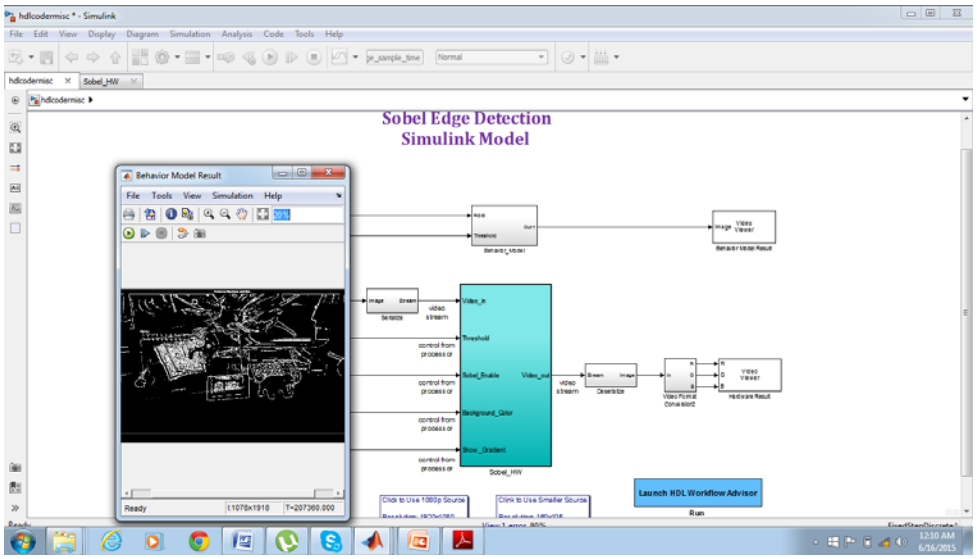
where a1 .. a9 are grey levels of each pixel in the filter window.

X = -1*a1 + 1*a3 - 2*a4 + 2*a6 - 1*a7 + 1*a9

Y = 1*a1 + 2*a2 + 1*a3 - 1*a7 - 2*a8 - 1*a9

Sobel Gradient = sqrt(X*X + Y*Y)

3.3 Simulink Model of Implemented Sobel Edge Detection



4. FPGA IN LOOP IMPLEMENTATION

4.1 Model-Based Design for Implementation

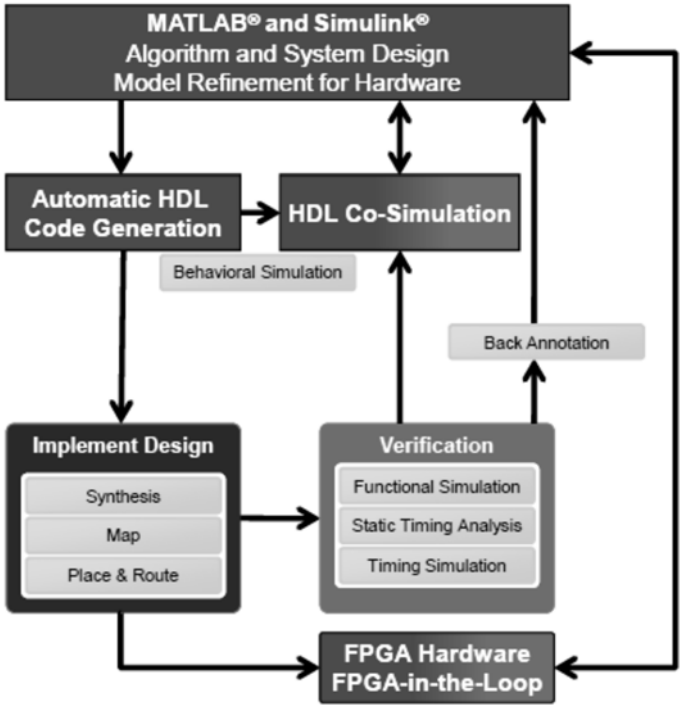


Fig 5: FPGA in Loop

FPGA-in-the-loop (FIL) simulation provides the capability to use Simulink or MATLAB software for testing designs in real hardware for any existing HDL code. The HDL code can be either manually written or software generated from a model subsystem. FIL performs the following processes when it creates the block or System object.

-Generates a FIL block or FIL System object that represents the HDL code.

-Provides synthesis, logical mapping, place-and-route (PAR), programming file generation, and communications channel. Loads the design onto an FPGA.

-As part of FIL simulation, the block or System object and your model or application, Transmits data from Simulink or MATLAB to the FPGA Receives data from the FPGA Exercises the design in a real environment. FIL provides the communication channel for sending and receiving data between Simulink and the FPGA. This channel uses a Gigabit Ethernet connection. Because communication between Simulink and the FPGA is strictly synchronized, the FIL simulation provides a more dependable verification method. FIL programming file, the software performs the following tasks:

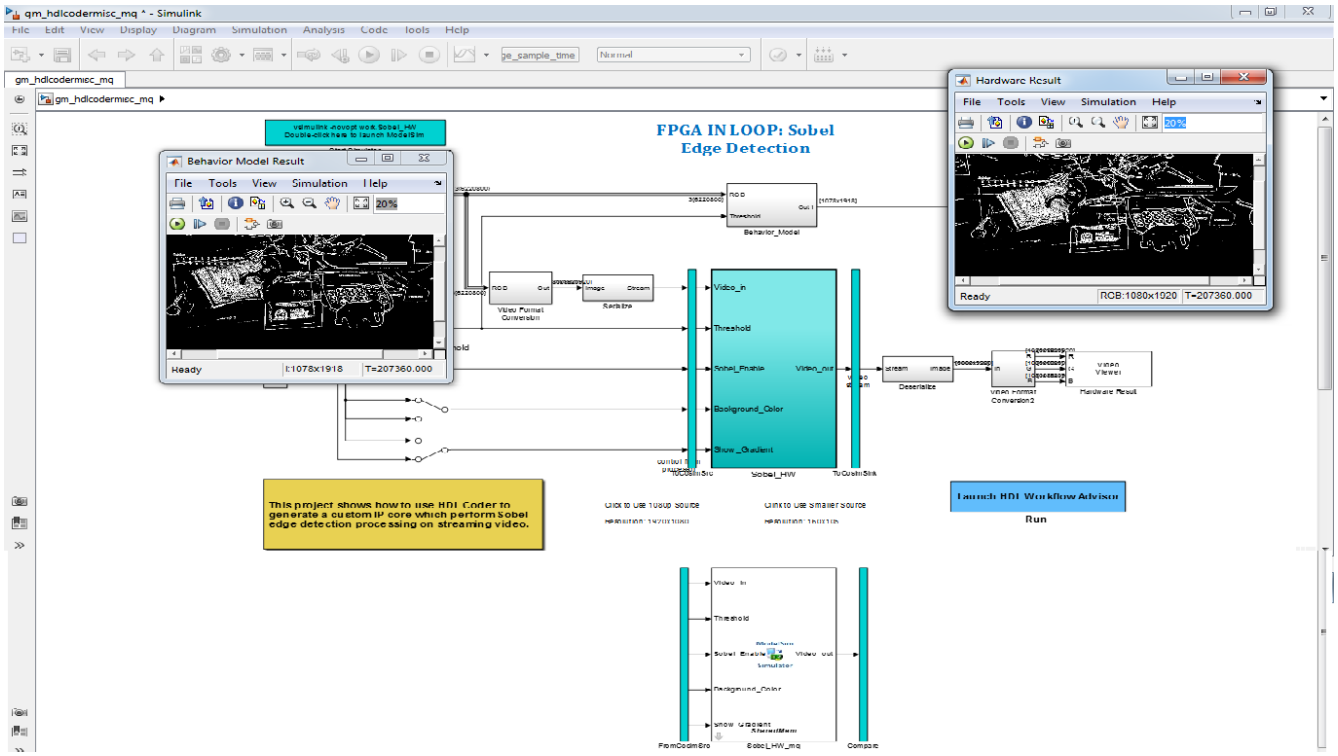
-Generates HDL code for the specified DUT and creates an ISE or Vivado project.

-Along with your FPGA design software, synthesizes, maps, places and routes, and creates a programming file for the FPGA.

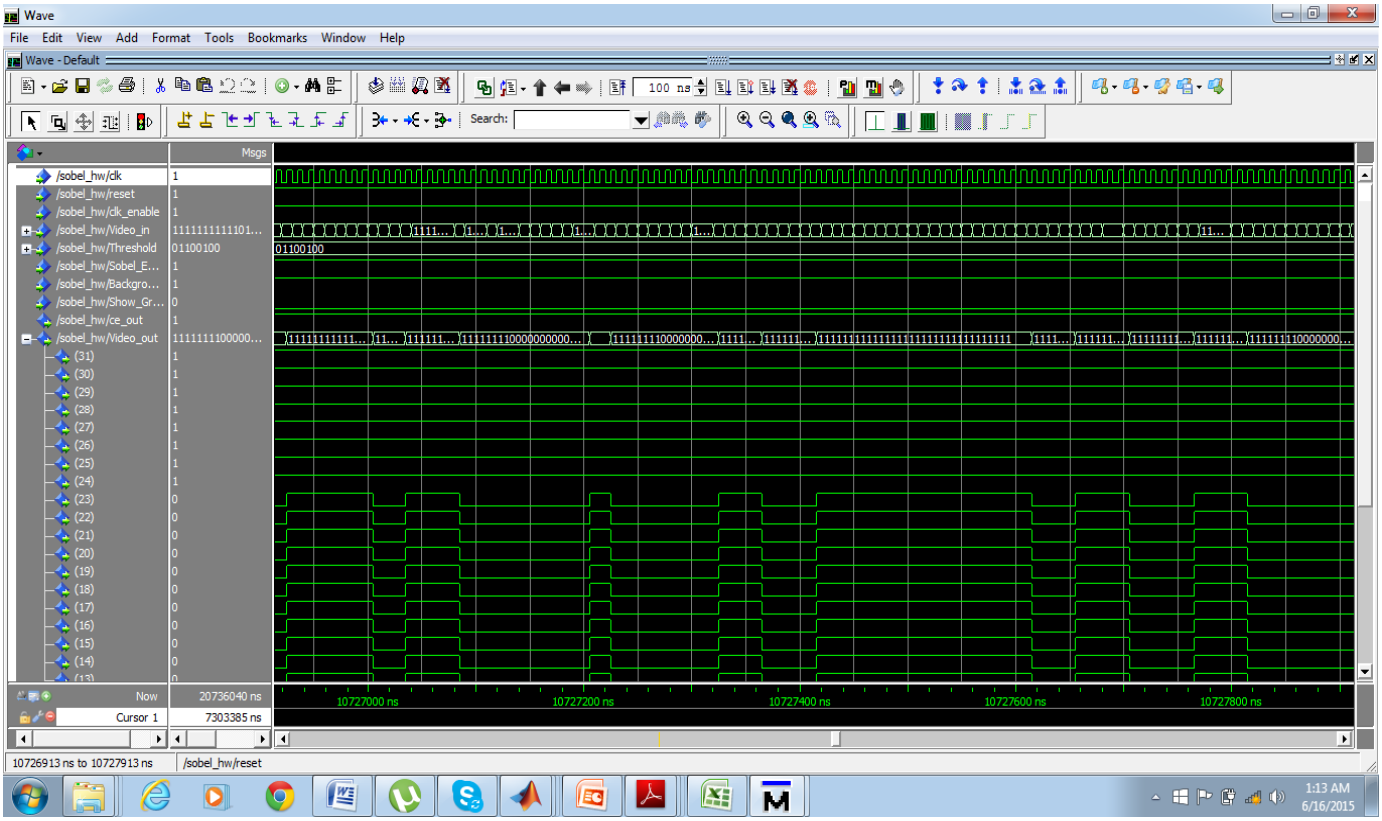
-Downloads the programming file to the FPGA on the development board through the board's normal configuration connection. Typically, that connection is a serial line over a USB cable.

5. Results

5.1 Generate RTL Code and Verification With HDL Co-Simulation of SE.

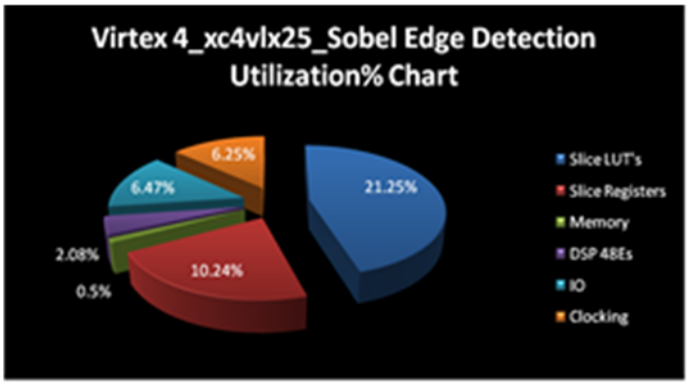


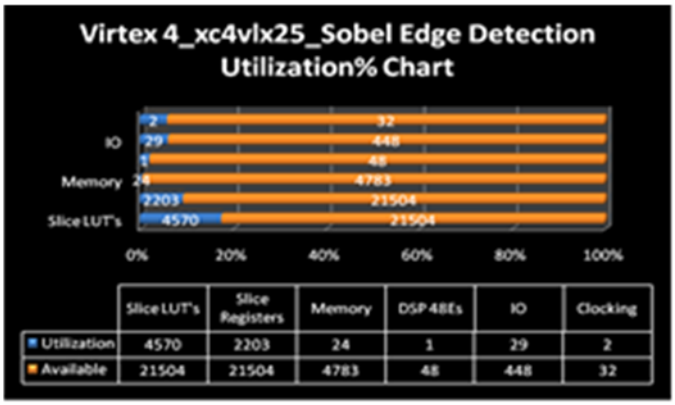
5.2 RTL Test Bench of Soble Edge Detection.



5.3 Comparative Analysis with Virtex 4

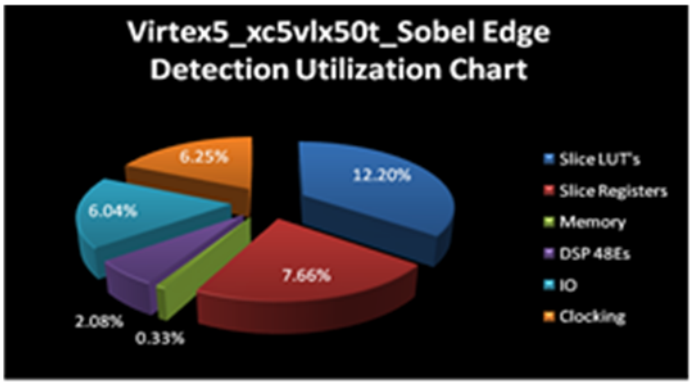
Resource	Utilization	Available	Utilization%
Slice LUT's	4570	21504	21.25
Slice Registers	2203	21504	10.24
Memory	24	4783	0.50
DSP 48Es	1	48	2.08
IO	29	448	6.47
Clocking	2	32	6.25

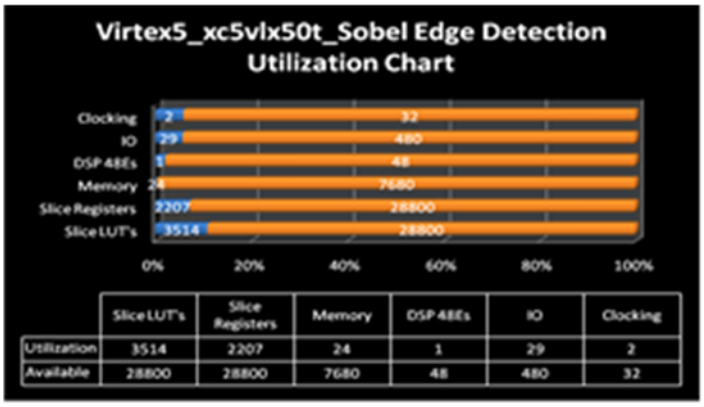




5.4 Comparative Analysis with Virtex 5

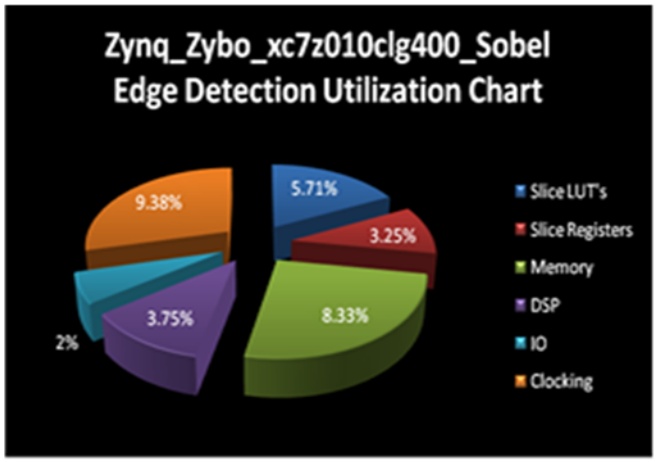
Resource	Utilization	Available	Utilization%
Slice LUT's	3514	28800	12.20
Slice Registers	2207	28800	7.66
Memory	24	7680	0.31
DSP 48Es	1	48	2.08
IO	29	480	6.04
Clocking	2	32	6.25

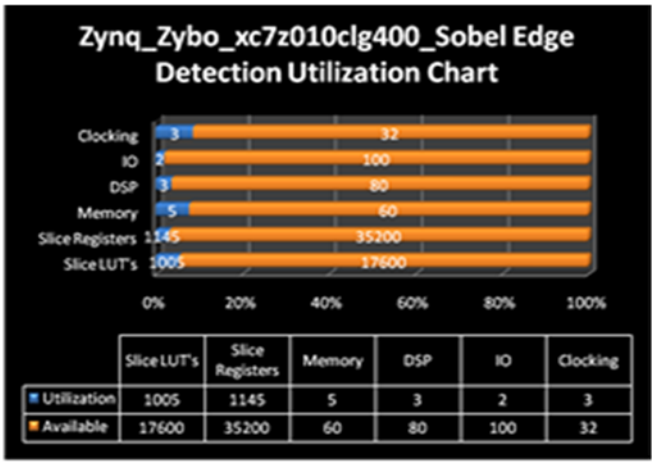




5.5 Comparative Analysis with Virtex 5

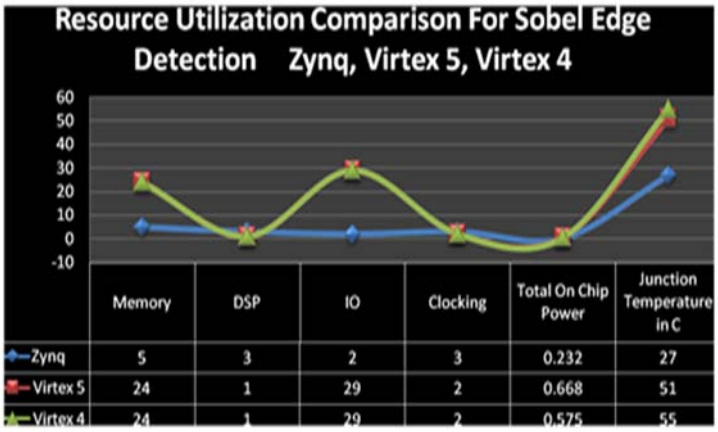
Resource	Utilization	Available	Utilization%
Slice LUT's	1005	17600	5.71
Slice Registers	1145	35200	3.25
Memory	5	60	8.33
DSP	3	80	3.75
IO	2	100	2.00
Clocking	3	32	9.38

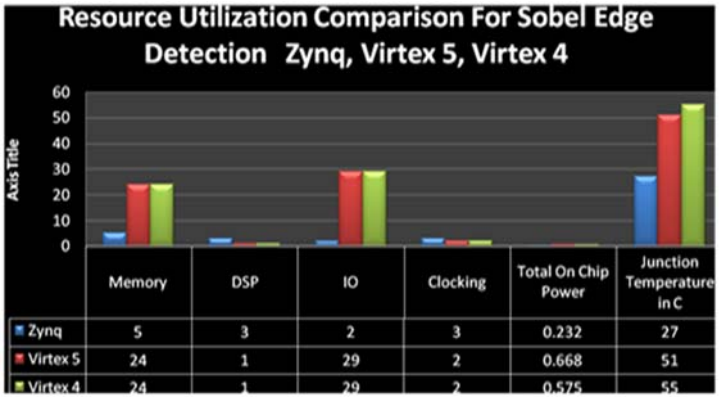




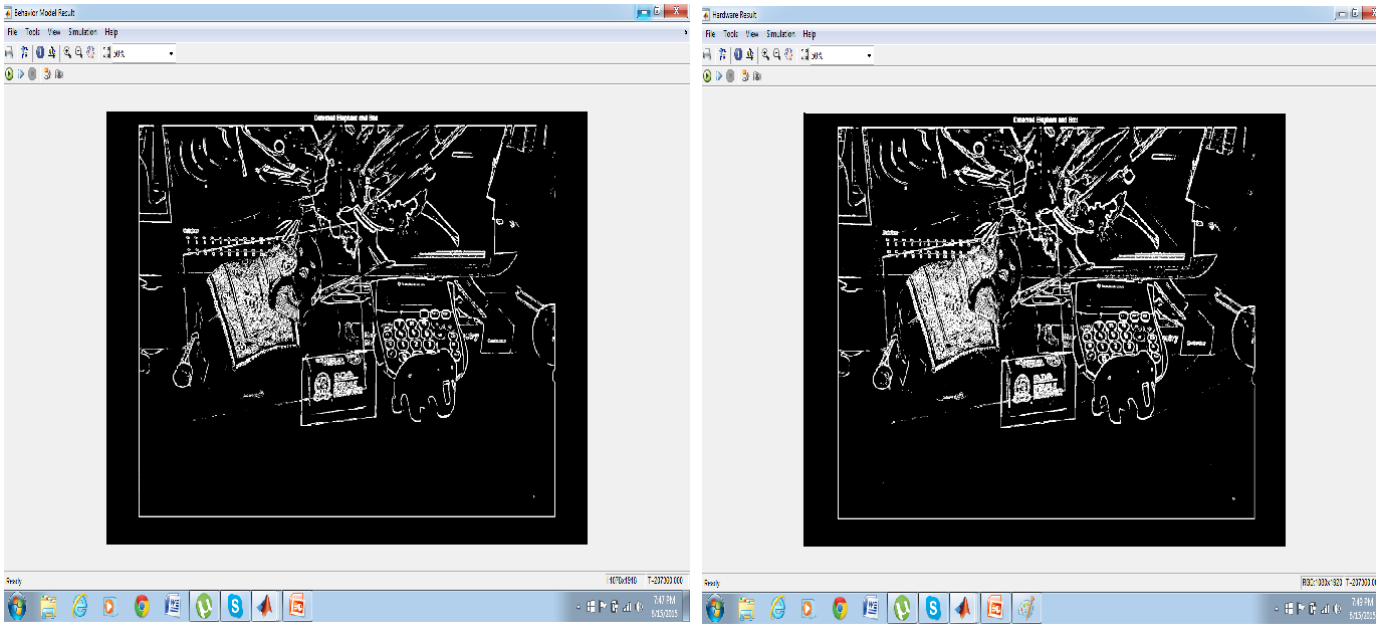
5.6 Resource Utilization Camparison

Resource	Zynq	Virtex 5	Virtex 4
Slice LUT's	1005	3514	4570
Slice Registers	1145	2207	2203
Memory	5	24	24
DSP	3	1	1
IO	2	29	29
Clocking	3	2	2
Total On Chip Power	0.232	0.668	0.575
Junction Temperature in C	27	51	55





Comparative analysis between Software and Hardware results



6. Conclusion

Implementation of a video processing algorithm on the FPGA is complex, tedious and error prone when using traditional design methodologies. Since time-to-market is very important, it is required to look at the product development cycle to reduce the design time and gain a competitive edge in the time to market. Hardware description of a real-time edge detection system based on VHDL was considered of high-speed processing in image edge detection. A hardware architecture of Sobel edge detection algorithm for implementing on field programmable gate array (FPGA) chips was proposed. The proposed architecture calculates the edges of gray scale images. Simulation results and synthesizing proposed Sobel edge detection processor on Xilinx Zynq Zybo SoC xc7z010clg400 FPGA chip demonstrated the efficiency of proposed architecture for edge detecting of gray scale 1080×1020 images for real-time image processing applications.

7. References

1. S. U. R. F. (SURF), “Herbet bay, andreas ess, tinne tuytelaars,luc van gool,” Elsevier preprint, 2008.
2. Z. Guo, W. Xu, and Z. Chai, “Image edge detection based on FPGA,” in Proceedings of the 9th International Symposium on Distributed Computing and Applications to Business, Engineering and Science, pp. 169–171, August 2010.
3. C.Moore, H.Devos, andD. Stroobandt, “Optimizing the FPGA memory design for a sobel edge detector,” in Proceedings of the 20th Annual Workshop on Circuits, Systems and Signal Processing, 2009.
4. Z. Shanshan and W. Xiaohong, “Vehicle Image Edge Detection Algorithm Hardware Implementation on FPGA,” IEEE International Conference on Automation and Logistics, Vol.14, pp.184-188, 2010.
5. Yahia Said, Taoufik Saidani, Fethi Smach and Mohamed Atri, “Real Time Hardware Co-simulation of Edge Detection for Video Processing Time”, Electrotechnical Conference (MELECON), 16th IEEE Mediterranean, Vol.25, pp.852-855, 2012.
6. www.zynqbook.com, Designing with Vivado High 76 Level Synthesis v1.2, May 2014.
7. The MathWorks Inc., MATLAB & Simulink, HDL Coder, www.mathworks.com
8. Xilinx Inc., www.Xilinx.com, Xilinx Vivado Design Suite,www.Xilinx.com
9. S. Behera, M. N. Mohanty, S. Patnaik, “A Comparative Analysis on Edge Detection of Colloid Cyst: A Medical Imaging Approach,” Soft Computing Techniques in Vision Science, Studies in omputational Intelligence, Springer, Volume 395, pp 63-85, 2012.
10. T. A. Abbasi and M.U. Abbasi, “A proposed FPGA based architecture for sobel edge detection operator”, J. of Active and Passive Electronic Devices, Vol. 2, pp. 271–277.
11. G. Martin and G. Smith, “High-level synthesis: Past, present, and future,” IEEE Design & Test of Computers, vol. 26, no. 4, pp. 18–25, 2009. WSEAS TRANSACTIONS on SIGNAL PROCESSING Yahia Said, Taoufik Saidani, Mohamed.
12. C. Moore, H. Devos, and D. Stroobandt, “Optimizing the FPG Amemory design for a sobel edge detector,” in Proceedings of the 20th Annual Workshop on Circuits, Systems and Signal Processing, 2009.
13. Wolf, W. 2004. FPGA-Based System Design. Englewood Cliffs, NJ: Prentice- Hall.
14. Wu, J., Sun, J. and Liu, W. 2010. Design and Implementation of Video Image edge Detection System Based on FPGA. In Proceedings of 3rd IEEE International Congress on Image and Signal Processing.