

Object Tracking IP using Vivado HLS and Xilinx SoC FPGA



Parag Rao, Aman Kumar, Dr. Charles Chang Choo

Department of Electrical Engineering, San Jose State University, San Jose, California 95192

Introduction

Feature plays a very important role in the area of developing autonomous motor vehicles. The objective of this project is to track the direction of motion of a host vehicle by extracting its features. Feature points of the input frame was calculated using the “Fast Corner” algorithm. Employment of motion estimation algorithm on the current frame against a previous frame on these extracted features points provides an accurate direction of motion.

Algorithm Design for Object Recognition

The Object Recognition consists of the following stages:

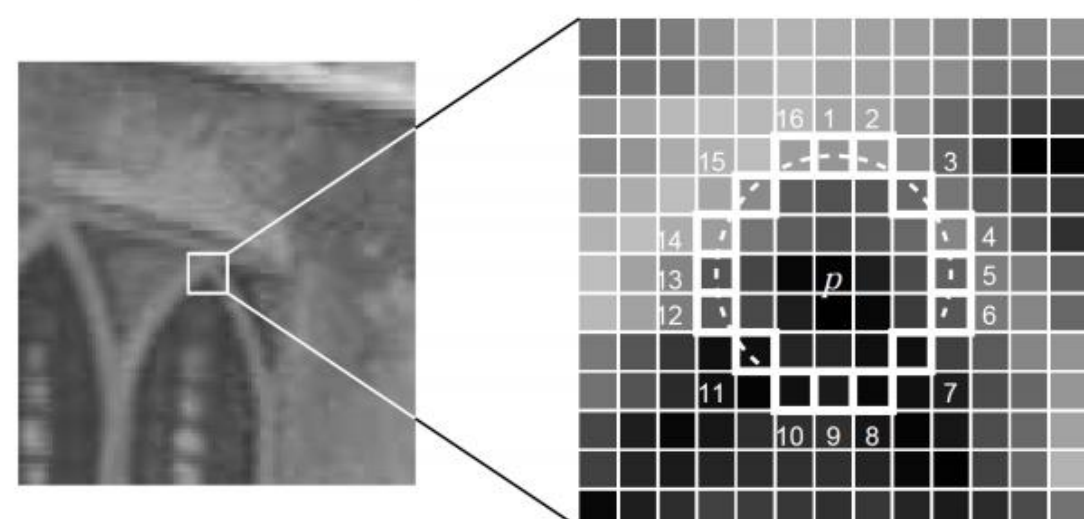
- Grayscale Conversion
- Delta Frame Generation
- Thresholding
- Noise Filtering
- Image Enhancement

Motion Estimation

In block matching algorithm, matching blocks in one frame are located in the successive frames of the digital video. The process involves division of current frame into number of macro blocks and then comparison of each corresponding block in adjacent frames. A vector can be used to find the movement of that particular block in the next frame or previous frame. We have used SAD as block matching algorithm. SAD is sum of absolute differences. SAD is the method to find the measurement of similarity between different image blocks.

Algorithm for Fast corner detection :

- Step 1. Select a pixel „p” in the image. Assume the intensity of this pixel to be IP
- Step 2. Set a threshold intensity value T,.
- Step 3. Consider a circle of 16 pixels surrounding the pixel p.
- Step 4. “N” contiguous pixels out of the 16 need to be either above or below IP by the value T, if the pixel needs to be detected as an interest point.
- Step 5. To make the algorithm fast, first compare the intensity of pixels 1, 5, 9 and 13 of the circle with IP. As evident from the figure below, at least three of these four pixels should satisfy the threshold criterion so that the interest point will exist.
- Step 6. If at least three of the four pixel values - I1 ,I5 ,I9 I13 are not above or below IP + T, then P is not an interest point (corner).
- Step 7. Repeat the procedure for all the pixels in the image



Design Approach in Vivado HLS

Step 1: Develop and execute an OpenCV application on Desktop.

Step 2: Recompile and execute the OpenCV application in the Zynq SoC without modification.

Step 3: Refactor OpenCV application using I/O functions to encapsulate an accelerator function.

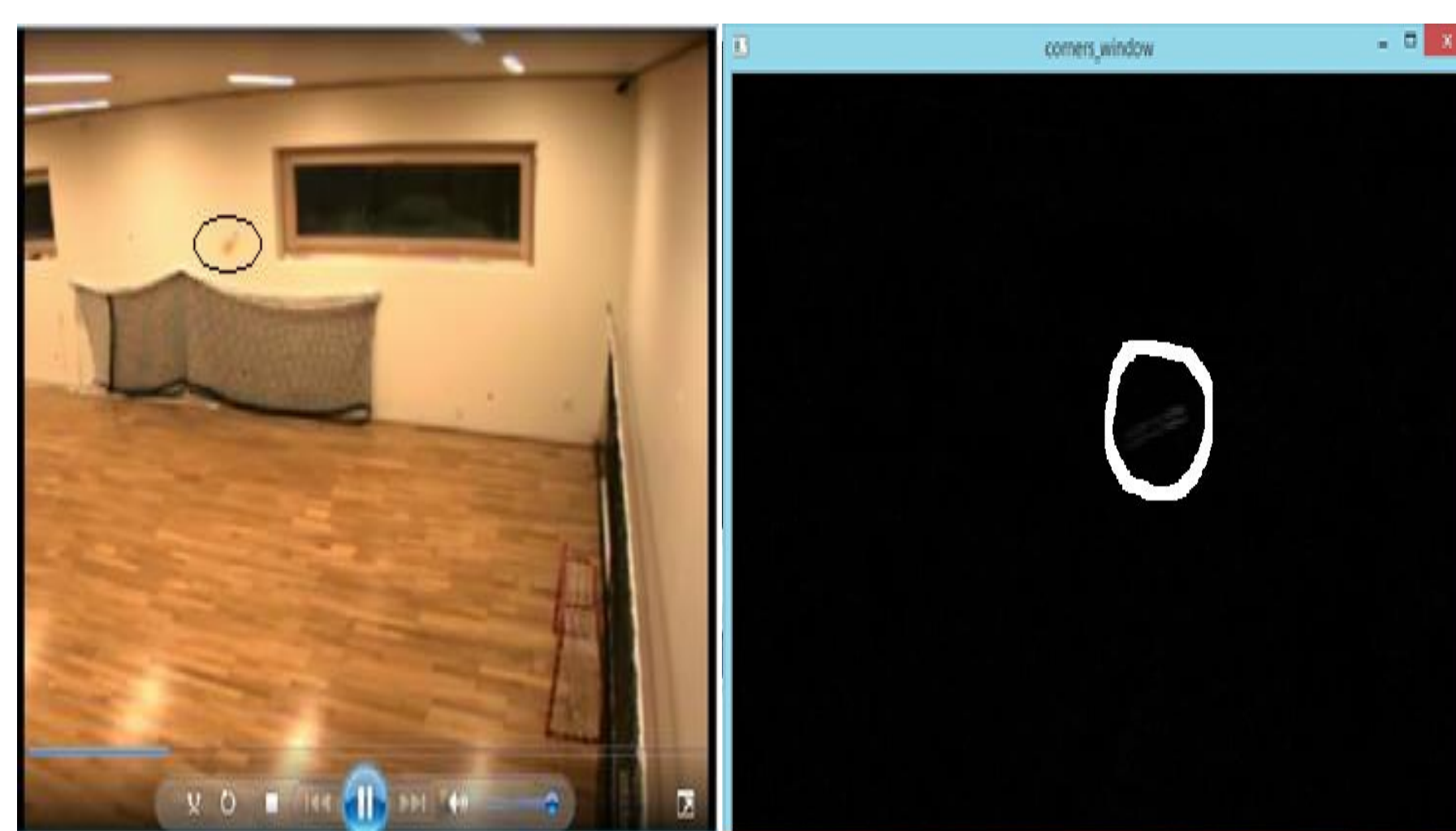
Step 4: Replace OpenCV function calls with synthesizable video library function calls in accelerator function.

Step 5: Generate an accelerator and corresponding API from the accelerator function using Vivado HLS.

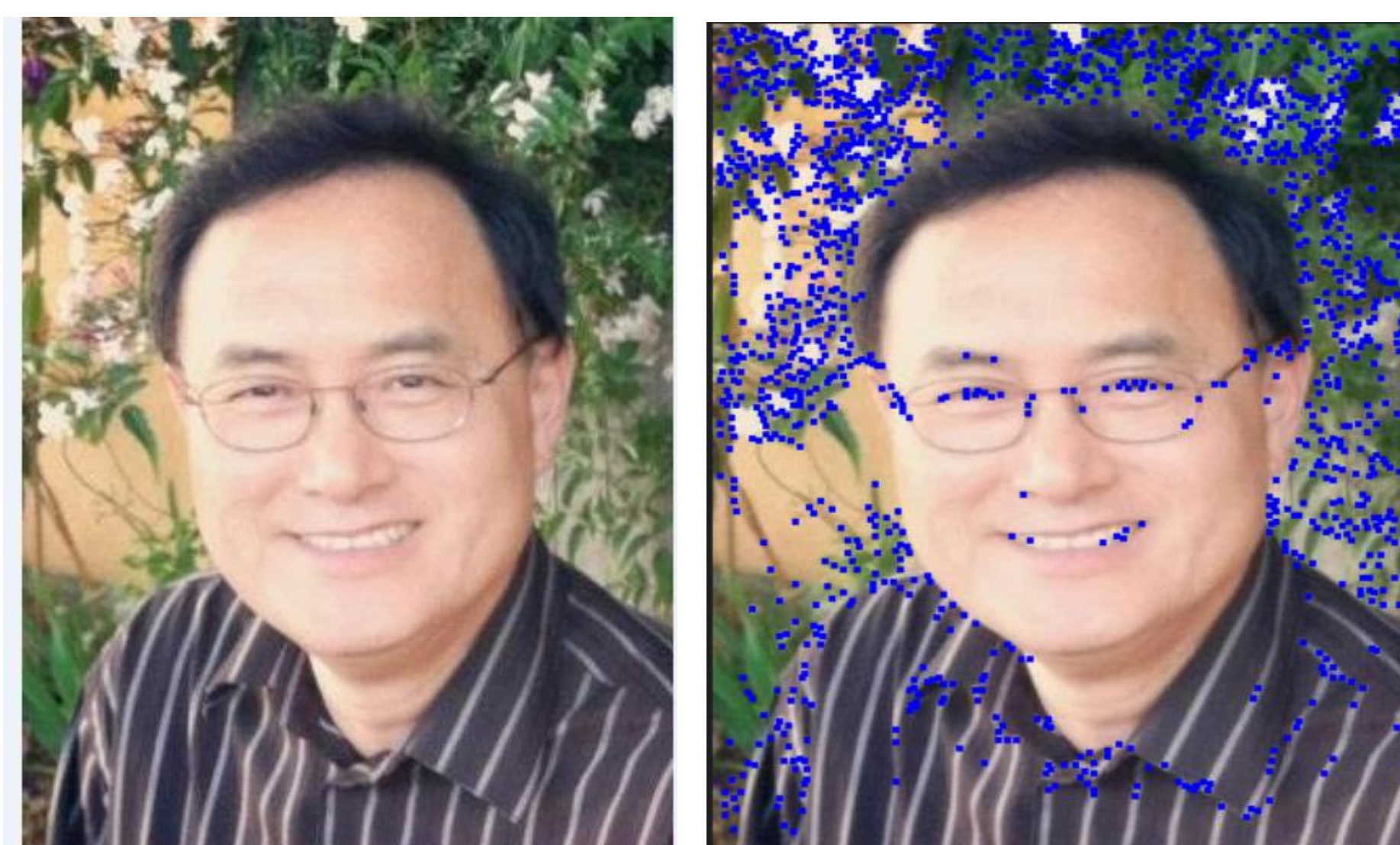
Step 6. Replace calls to the accelerator function with calls to the accelerator API

Step 7. Recompile and execute the accelerated application [1]

Results



The above image shows the object being tracked using OpenCV. It takes a video of a bouncing ball as an input and captures one frame. The algorithm then converts the image into a grayscale image. Once this conversion is done, the algorithm captures another image and an absolute sum of difference of these two images is computed. The resulting image is as displayed in the frame above on right side.

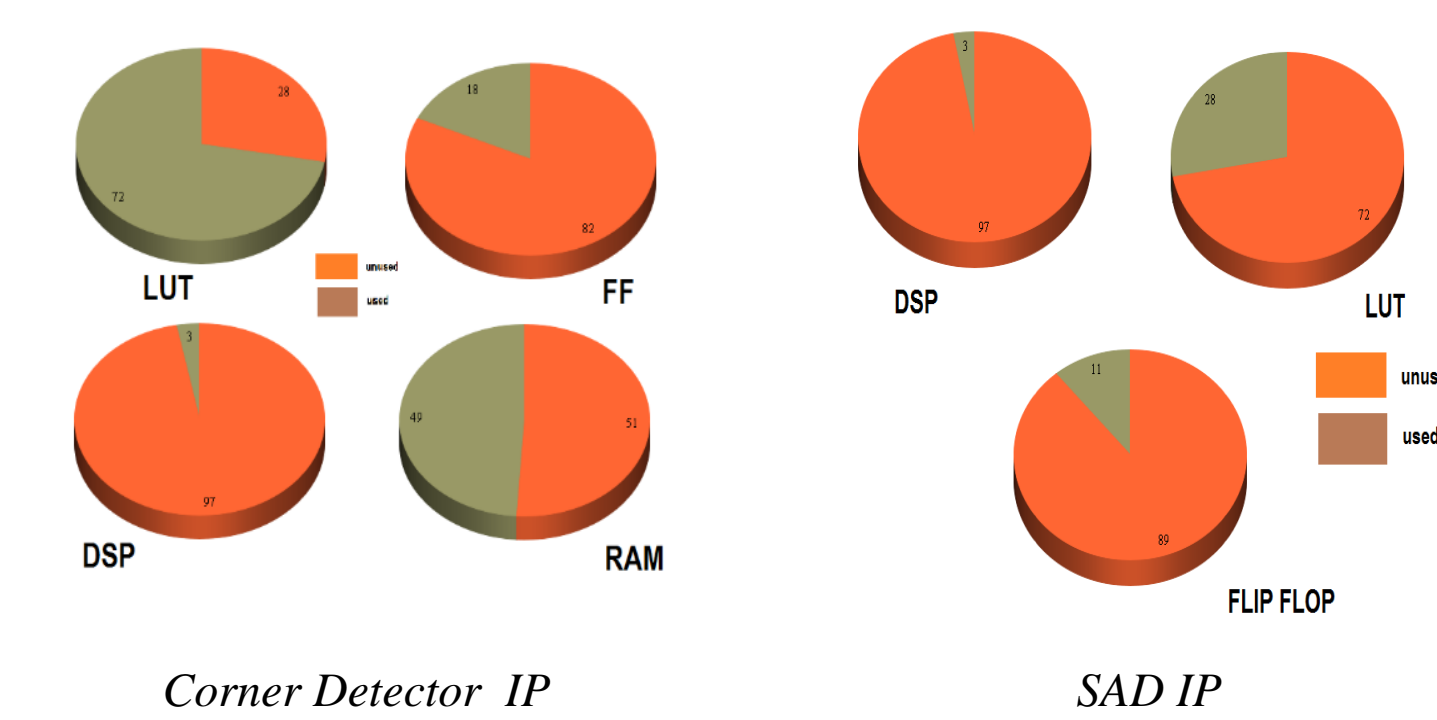


The above images show corners detected on two separate images. This image was then compared against the golden reference model to see if the synthesizable libraries were able to convert the image successfully

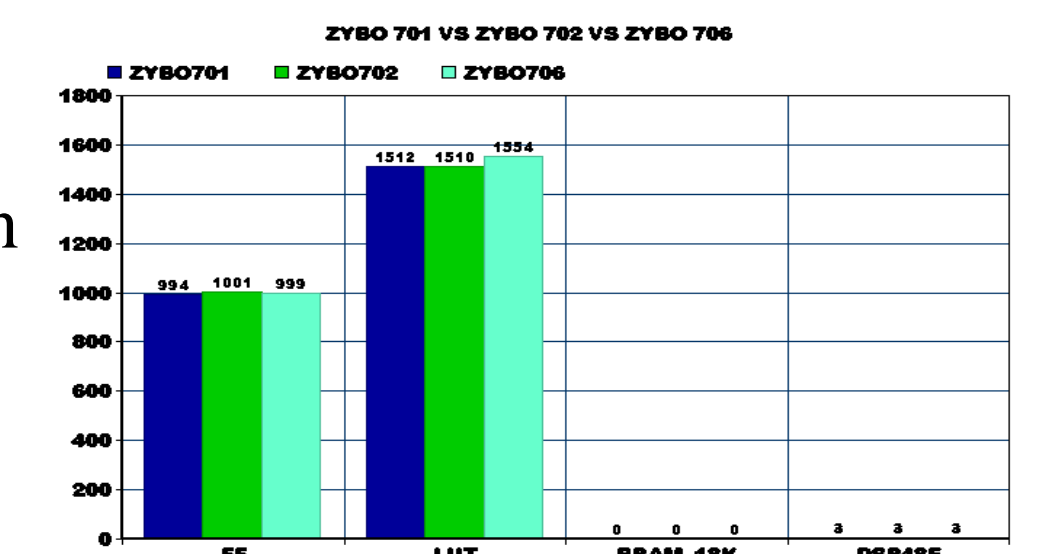
Analysis

Resource Utilization

The resource utilization of Xilinx Zybo 701 board

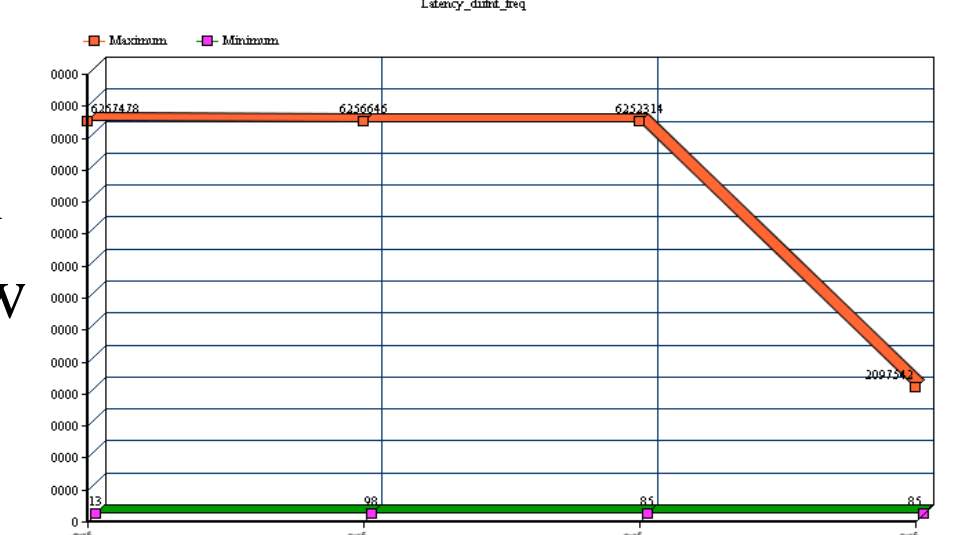


Comparison of resource consumption of three ZYBO SoC boards (Z7010,Z702,Z706)

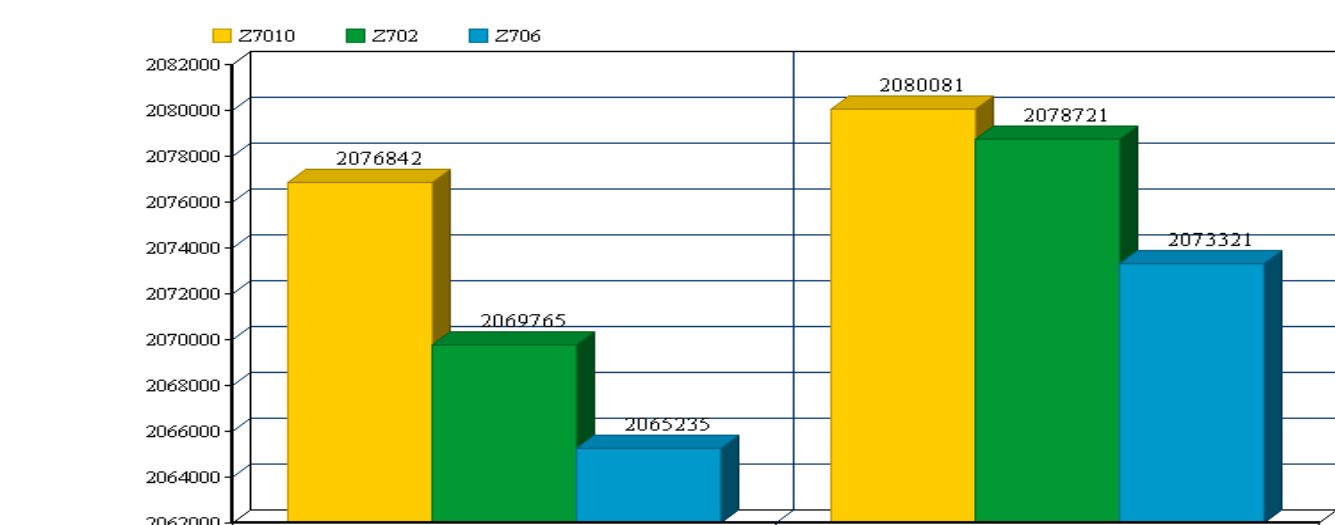
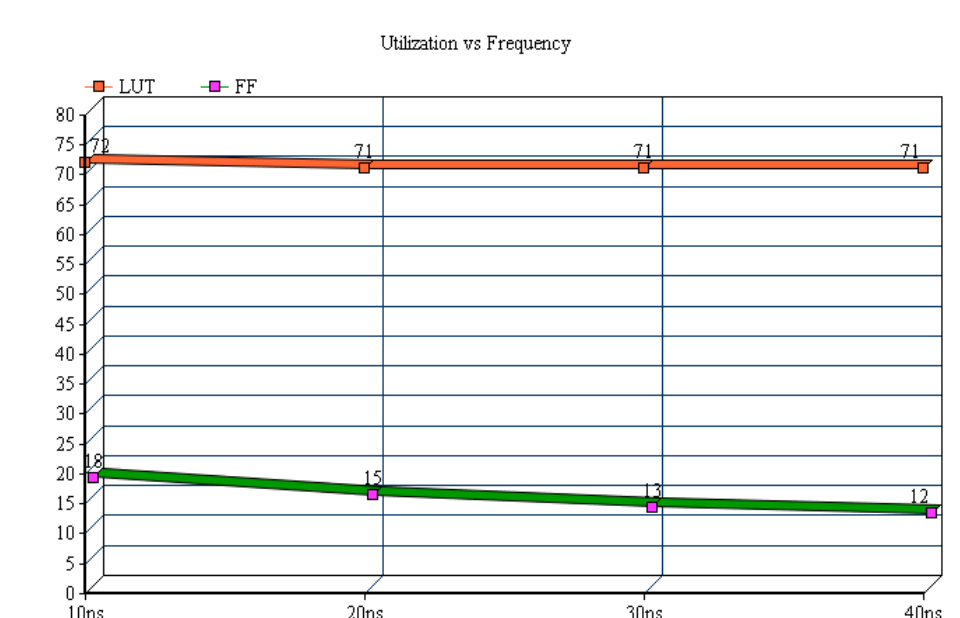


Timing Analysis for Z7010

Latency decreases when We run our design below 33.3MHz(30ns)



Percentage of flipflops utilization goes on decreasing. However, the percentage of LUT's remains same.



Minimum and maximum latency for ZYBO SoC boards (Z7010,Z702,Z706)

Conclusions And Future Scope

We successfully tracked a bouncing ball on OpenCV and emulated it on Vivado HLS. This IP was then converted into synthesizable code and an RTL based IP was generated. Resource utilization of our corner detection IP consumes 3 DSP blocks, 5489 flipflops, and 12558 LUT's. The motion estimation IP consumed 994 flipflops, 1512 LUT's, 6 DSP blocks. The design ran successfully on 100MHz with an uncertainty of 1.25ns. Our design can process ten frames per second. Latency decreases when We run our design below 33.3MHz(30ns)

Our future work includes the exploration of more robust models in order to track objects with complex textures. We also plan to implement the same algorithm on GPU for better performance. GPU are very efficient at manipulating image processing, and it's highly parallel structure makes them more efficient than FPGA.

Key References

- [1] Xilinx HLS Synthesis and simulation design guide <http://www.xilinx.com/itp/xilinx10/books/docs/sim/sim.pdf>
- [2] Ismail Haritaoglu, David Harwood, and Larry S. Davis. “W4: Real-Time Surveillance of People and Their Activities”. IEEE Transactions on Pattern Analysis and Machine Intelligence. Pages 809-830. 2000.
- [3] L. Wixson. “Detecting Salient Motion by Accumulating Directionally-Consistent Flow”. IEEE Transactions on Pattern Analysis and Machine Intelligence. Pages 774-780. April 2000.

Acknowledgments

We thank Dr. Charles Chang Choo for providing invaluable expertise, laboratory space and equipment.

For further information

Please contact parag.rao@sjsu.edu or aman.kumar@sjsu.edu for HLS IP, Vivado simulation files, OpenCV code and synthesis reports are available upon request.