Real Time Optical Flow System Design On FPGA

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

Introduction

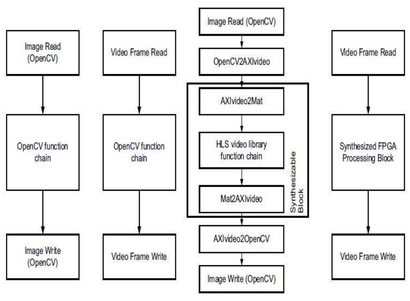
This paper shows an extensively used algorithms i.e. FAST (Features from Accelerated Segment Test) and SAD (Sum of Absolute Difference) being developed on a Field Programmable Gate Array to detect motion of any moving object in real time.[12] Detection of any moving object is one of the important tasks in driverless cars. By detecting moving objects around the car, decisions on speed, direction of the car etc. can be computed.

Two dimensional image motion is concluded from the objects with three dimensional motion. On the same image plane, motion is related to a visual sensor. We can estimate the motion of projected two dimensional image in terms of velocities of instantaneous image or displacements of discrete image allowed by time ordered image sequences. These are commonly known as image velocity field or optical flow field. If the optical flow is an approximation of two dimensional image motion, then it can be used to recover the three dimensional motion of sensor.

Optical flow computation of any real time moving object can be done by various algorithms. There are various new techniques which are being developed. SAD and FAST are hugely used algorithms used for this implementation. In this project, these algorithms were implemented on the field programmable gate array. By doing this, optical flow of any object in real time can be found. Optical flow shows how the object is moving around and distance from object. The system works on the HD video of 1980x1080 resolution.

Architecture

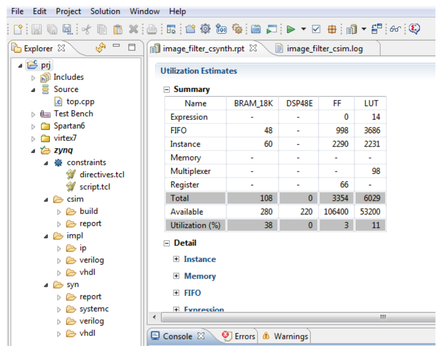
Software Implementation
Using the Vivado HLS video libraries, we can implement OpenCV applications on ZC702. From prototyping of algorithm to execution in system, in this design process OpenCV can be used many a times.[2] With the help of Vivado HLS we can get synthesizable C++ code using video libraries.



Hardware Implementation
Xilinx is one of the renowned companies making field programmable gate array (FPGA) boards. In this project Xilinx Zynq ZC702 board was used. This board comes up with many useful features that we can use in designing systems.

Design Flow

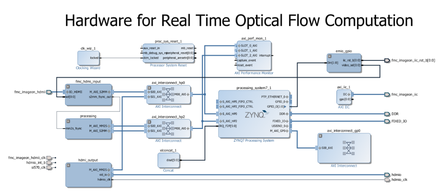
We will now see the steps to develop hardware environment for the system design using the Xilinx Vivado Design Suite, Vivado HLS, XSDK (Software Development Kit and PetaLinux design tools). That will focus on the design flow specific for the Zynq architecture. The design was built on the TRD (Targeted Reference Design) base of the ZC702.



For the migration of algorithms which are coded in C, C++ to RTL code, Vivado HLS is useful tool and environment for this migration. Hardware will run on HDL (Hardware Description Language) which is generated by Vivado in terms of bit stream. Below is the screenshot for the HLS project built in Vivado HLS and generated synthesis report of that project. SAD and FAST algorithms were written in C.

These algorithms were simulated successfully and then synthesized successfully in Vivado HLS. HD video is captured and then converted into subsequent frames for the implementation of FAST and SAD algorithm. By this synthesis we can get data like utilization estimates in that, all the utilizations of memory (BRAM), slices (DSP), flip flops (FF), look up tables (LUT). You can see spartan6, virtex7 and zynq in the left side menu i.e. for the comparison of the results from three different families.

Hardware for Real Time Optical Flow Computation



The above figure is the schematic for the hardware implementation of real time optical flow computation. There are various blocks connected to Zynq processing system. The connected blocks are clocking wizard, processor system reset, AXI performance monitor, HDMI output and input and AXI interconnect. In the above block diagram you can see that processing system consists of memory interfaces, input output peripherals, application processor unit and interconnect. Whereas programmable logic consists of common and custom peripherals and accelerators.

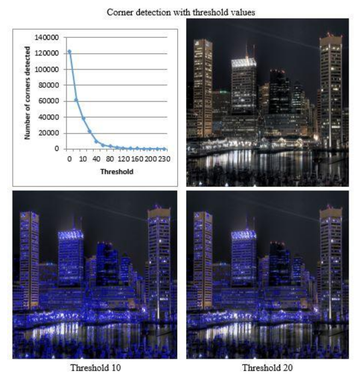
Analysis & Results

As to summarize the results, when only SAD algorithm is applied on the frame of image, there is an unwanted noise noticed in the image. After applying FAST, the frame of image was obtained in corners of the object. Now if we apply SAD on this corner points, all the unwanted noise was eliminated from the frame of image. The condition to calculate SAD in each block was that if there is at least one corner detected in the block.

Algorithm Comparison

Image Size 1080p	Harris	FAST
Target Clock	6.67	6.67
Estimated Clock	6.63	6.11
BRAM_18K	125	59
FF	19923	7667
LUT	31174	13112
min (ns)	483	174
max (ms)	6.2359	6.2848

From the figure above we can see the comparison between Corner Harris and FAST algorithms. Corner Harris algorithm was implemented with large numbers of FF, LUT as compared to FAST algorithm. FAST is preferable over Harris. Both the algorithms implemented on the same image size of 1080p of resolution. The figure below shows for the different threshold value the numbers of corners detected varies. With the increment in threshold value, the numbers of corners detected decreases as you can see in images above with the original image given.



Conclusions & Future Work

Finally to summarize this project, conclusion can be made that a new technique was developed in this project to detect motion and evaluate optical flow of any object. By implementing FAST and then SAD algorithm, complex calculations can be reduced and performance can be more efficient than other algorithms.
As to extend this project further one can pipeline SAD and FAST algorithms on the hardware to make system more efficient to eliminate complex calculations. Implementation of optical flow can be done on Virtex 7 board. This will make system perform more efficiently than Spartan 6 and Zynq.

Key References

[1]K. Toyama, J. Krumm, B. Brumitt, and B. Meyers, “Wallflower: principles and Practice of background maintenance” in Proc. 7th IEEE Conf. Computer Vision, 1999, vol. 1, pp. 255–261.
[2]“http://www.xilinx.com/support/documentation/boards_and_kits/zc702_zvik/ug850-zc702-eval-bd.pdf “.
[3]G.L.Foresti,”A Real Time System for Video Surveillance of Unattended Outdoor Environments”.

[4]J. M. Ferryman, Ed., in Proc. 9th IEEE Int. “Workshop on Performance Evaluation of Tracking and Surveillance”, 2006.

Acknowledgments

I would acknowledge my project advisor, Prof. Morris Jones for all his teaching and guideline throughout the project. Without him I could not complete this project.

For further information

Please contact *vishalprajapati_0012@yahoo.com*

408-614-8841