Design and Implementation of SIFT Algorithm on FPGA for Autonomous Vehicle Vision Applications

Arjun V Kandukuri, Viswateja Nemani , Dr. Chang Choo

Department of Electrical Engineering, San Jose State university, San Jose, California 95192.

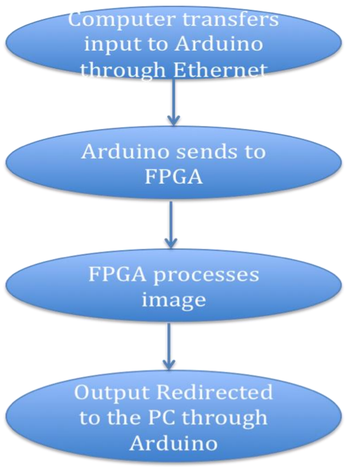
Introduction

The main objective of this project is to design a SIFT engine on FPGA by exploiting the speed and parallelism in hardware, an engine immune to invariance of scale, illumination and orientation, an engine powerful enough for real time video processing, efficient enough to consume less resources and power, along with ease of integrating this in designs by other users.

SIFT is used a base for many computer vision applications and our objective is to create a robust SIFT implementation algorithm which can improve the speed and efficiency of other designs that use this engine. The design includes applying Gaussian function to Image, calculation Difference of Gaussian, a Harris Corner algorithm to eliminate edges, local extrema determination and finally finding the keypoints and keypoint descriptors. These keypoints are immune to scale, illumination and rotation. The keypoints can be used for various image-matching applications.

Implementation

SYSTEM FLOW



Implementation

HARDWARE MODULE

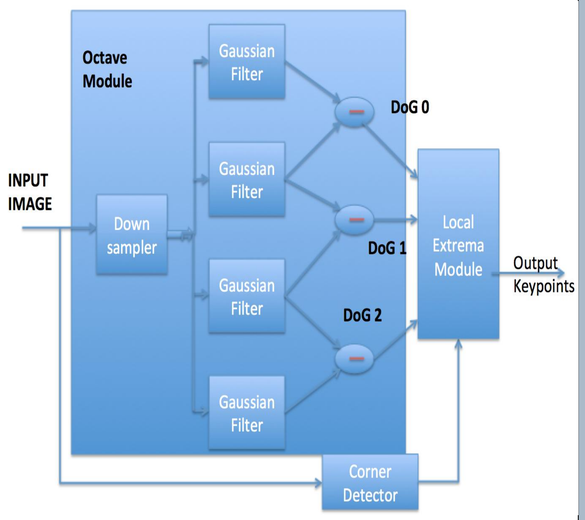
This paper discusses hardware implementation of the SIFT algorithm on an FPGA. This computer vision algorithm does a key point extraction and detection to process an image and extract local feature vectors that are immune to any scaling, rotation or translation of the image. Methods to accelerated hardware solution on FPGA to process images using SIFT are described. Also, hardware modules for the Down-samplers, Gaussian-filters and for computation of difference of Gaussians are developed to aid in expediting the process of this algorithm. The RTL is developed in Verilog, tool used is Xilinx ISE-14.4 and targeted device is Xilinx FPGA-Spartan6.

The Top module contains the following main Sub Modules:

1) Octave Module (Difference of Gaussians + Gaussian Functions)

2) Harris Corner Detector Module

3) Local Extrema Module



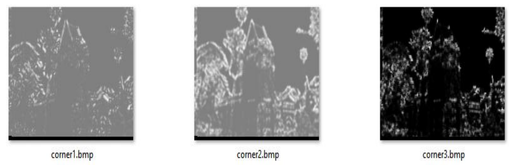
Results

This section describes the image simulation flow of the system.

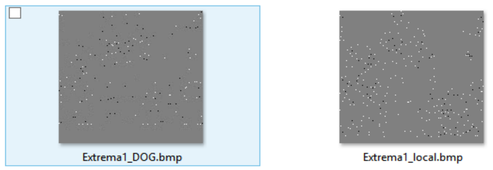
1) Input Image

2) DOG Output

3) Harris Corner Output

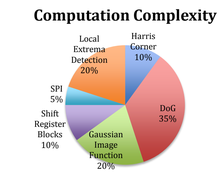


4) Local Extrema Output



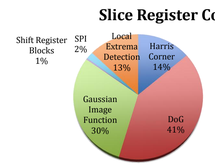
- The input image is DownSampled and applied to different Gaussian scales for Difference of Gaussians (DOG) images in (2).
- The Harris Corner module uses this same image as a reference to show the corners in (3).
- The Local extrema module as seen in 4) gives us the Minima and Maxima of the images in 1) and 2). All values in the image are scaled to either 255 or 0.
- Keypoints are then described by comparing the difference of the 2 pictures in 4).

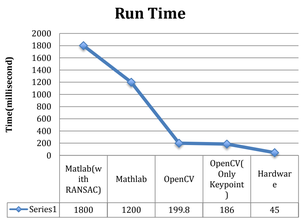
Analysis



According to the chart, the DOG and Gaussian functions requires more than 50% of total computation.

According to the chart, the DOG module utilizes most of the slice registers in FPGA.





From the graph, we conclude that the hardware takes less execution time compared to Matlab and OpenCV.

Summary

Successful Implementation of the SIFT algorithm on an FPGA has proved to be faster than software based approaches. As expected the DOG module has consumed almost 50% resources. A DOG module as a hardware accelerator along with a software part for the SIFT matching could prove to be a beneficial method. The Algorithm requires approximately 100k-LUTs and 1Mb on-chip memory.

Key References

[1] D. G. Lowe, “Distinctive Image Features from Scale-Invariant Keypoints,” IJCV, 2004.

[2] YinanYu ,Kaiqi Huang, Wei Chen and Tieniu Tan “A Novel Algorithm for View and Illumination Invariant Image Matching” , IEEE TRANSACTIONS ON IMAGE PROCESSING, VOL. 21, NO. 1, JANUARY 2012

Acknowledgements

We would like to thank Prof. Chang Charles Choo for his unending support and guidance. He has been a corner stone throughout the project. We would also like to thank Prof. Thuy Le and Prof. Morris Jones for their advice. Finally we would like to thank Xilinx University Program and DSP/FPGA Lab at San Jose State University for providing us the resources needed to complete the project.