GESTURE RECOGNITION SYSTEM USING KINECT CAMERA IMPLEMENTED ON FPGA

Vidhyapathi C. M. and Alex Noel Joseph Raj
School of Electronics Engineering VIT University, Vellore, Tamil Nadu, India
E-Mail: vidhyapathi.cm@vit.ac.in

ABSTRACT

Kinect Camera based gesture recognition is currently playing vital role in image processing field for e.g. gesture based Xbox games. FPGAs now a day are the most versatile Programmable Logic Device. FPGAs can be used for numerous applications. This paper presents information about performing image processing algorithm using Microsoft Kinect Camera using Field Programmable Gate Arrays (FPGA). The Image Processing on FPGA is quite a complicated task because it requires different architecture in order to process the image. To facilitate processing of image on FPGA, MATLAB and Simulink along with Xilinx System Generator (XSG) tools is the easiest and most efficient way. Such tools convert the image into suitable formats that are supported by FPGA.

Keywords: MATLAB, smulink, xilinx system generator, XSG, image processing, FPGA, kinect camera, hardware co-simulation.

INTRODUCTION

Over the past few decades, the field of image processing has undergone a rapid evolution; Gesture Recognition system is one such example of it. Gesture Recognition using Kinect Camera facilitates us to extract desired information from an image or a video, using a suitable algorithm. Implementing such algorithms using Kinect Camera along with FPGA to perform gesture recognition, is a faster and more accurate approach.

This paper provides a significant approach of gesture recognition using Xilinx System Generator (XSG) tools along with MATLAB and Simulink. Xilinx System Generator (XSG) is a tool which is capable of generating VHDL or Verilog code of the algorithm designed in Simulink. The generated code can be dumped into FPGA and desired Gesture can be recognized. The Image can be enhanced by creating a suitable algorithm using Xilinx system Generator such as Contrast Stretching, Image Filtering, Edge Detection, Thresholding. etc.
This paper aims at:

- Acquiring Image from Kinect Camera.
- Implementing gesture recognition algorithm on MATLAB using Xilinx System Generator (XSG).
- Software Simulation and Generation of VHDL code using Xilinx System Generator Token.
- Hardware Implementation of the given algorithm on FPGA board.

The Image is acquired using Microsoft Kinect Camera. The Gesture recognition algorithm is simulated using Xilinx ISE 14.3 along with MATLAB 2012a. The algorithm is implemented on FPGA Virtex-5 (XC5VLX50T) board.

DESIGN AND DESCRIPTION

A. Image acquisition

Image to be operated is taken from Kinect Camera using VU-Kinect available in mathworks.com. The block gives the RGB and Depth which can be directly operated in Simulink. The interested depth image can be acquired for the further processing.

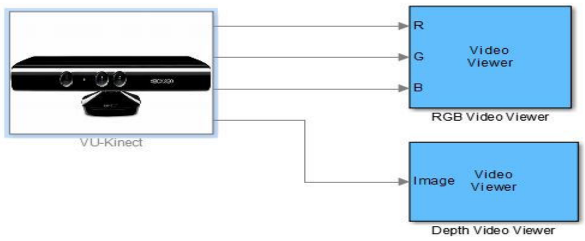


Figure-1. Acquiring image using Kinect Camera in Simulink.

B. Pre-Processing

The Image from Kinect cannot be directly fed to the FPGA board. It is necessary to pre-process the image and convert the image to serial form. The image obtained from Kinect Camera consists of two colors. The Closer Part is shown in blue color while the rest is with different color. Thus in order to enhance the image, the obtained image is converted to Y-Cb-Cr color space.

$$\begin{bmatrix} Y \\ Cb \\ Cr \end{bmatrix} = \begin{bmatrix} 0.299 & 0.587 & 0.114 \\ -0.147 & -0.289 & 0.436 \\ 0.615 & -0.515 & -0.100 \end{bmatrix} \begin{bmatrix} R \\ G \\ B \end{bmatrix}$$

After the conversion, the new enhanced image thus obtained is then transposed and then the transposed matrix is then converted to 1-D. This data is then converted to frames and then is passed to get unbuffered so that each matrix row becomes an independent time-sample in the output. This is done in order to adjust the sample period same at both the input and the output. The Pre-processing block set:

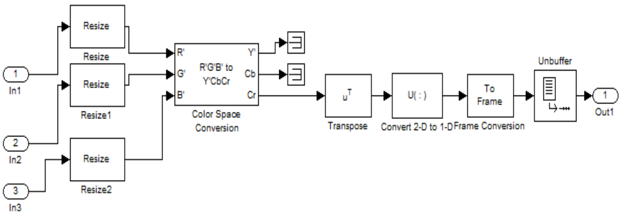


Figure-2. Pre-Processing.

C. Processing using Xilinx block sets

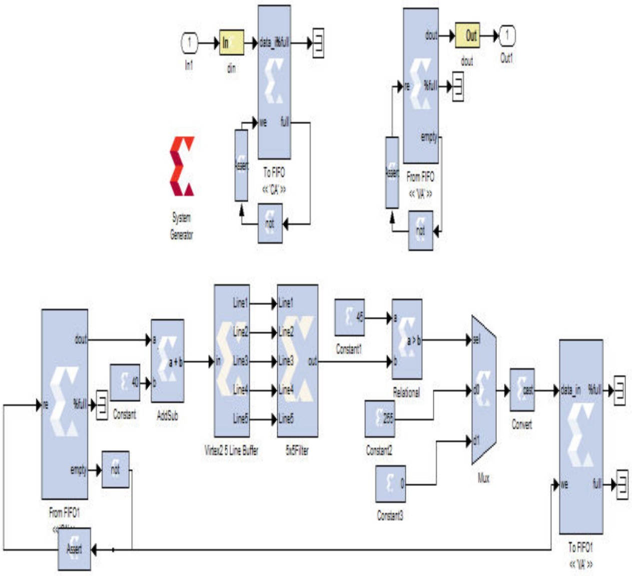


Figure-3. Edge detection using Xilinx blocksets.

The basic of edge detection requires sudden change in pixel values between the adjacent pixels. But sometime one may get a smoother image rather than having sudden pixel change. Thus in order to extract edges out of that kind of image the pixels thus obtained is enhanced using addsub block.

The 'Gateway in' block receives the input for the FPGA from the XSG in the form of serial data consisting of pixels. This data is then passed to FIFO blocks. When data is written to the software 'To FIFO' block during simulation, the same data is written to the FIFO in hardware. The design in hardware may then retrieve the data by reading from the FIFO. Similarly, when data is written into the hardware FIFO by design logic, the data may be read by the 'From FIFO' software block.

The data is then fed to the Virtex2-5 line buffer and this data is then transferred to the Xilinx filter blocks which perform its task of Sobel x-y edge detection. The Edge detected data is then passed over to get threshold. Thresholding is achieved using Xilinx 'Relational' and 'Mux' block set. The Output from the filter is a serial data containing range of values from 0 to 255. Thus, relational block is used to pass the '0' if the data is greater than 45 or else pass '1'. In order to get an enhanced pitch black and pitch white image, a 2x1 MUX is used. If the output of relational block is '0' then '255' is passed or else '0' is passed. This data is then passed to the 'To FIFO (VA)' block whose data is then read by 'From FIFO (VA)' and then passed out to the Simulink through 'Gateway out' block.

D. Post- Processing

The Data obtained after the FPGA processing needs to be converted back into the image. This is done with the post-processing subsystem.

The pixels obtained after getting processed are first converted back into their original data type using 'Data type Conversion' block available in Simulink library. The input data obtained can be converted to various data types using this block set.

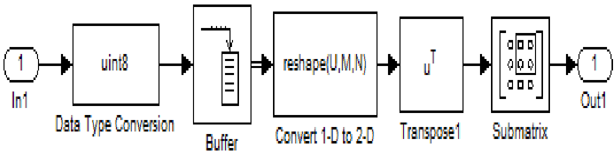


Figure-4. Post processing.

The output of the above mentioned block is serial in nature. This is passed to a block called 'Buffer' which acts as anti unbuffer block. The 'Buffer' block converts the high sampled serial data thus obtained into Frames.

After getting the Frames, these Frames are passed serially to 'Reshape' block available in Simulink Library. This Reshape block converts the 1-D data to 2-D Matrix. The number of columns and Rows are supposed to feed into the block to get desired numbers of Rows and Columns. After getting the 2-D Matrix the corresponding matrix is passed to 'Transpose' block which provides the same dimension to the matrix as that of the input image. The 'Submatrix' block set is used to reshape the image in order to obtain a better output.

E. Gesture recognition

After obtaining the Image (or Matrix) from the 'Transpose' block, the image is passed to the Gesture recognition Subset. This Subset consists of various other Simulink block which performs the task of comparing the gestures and displaying the corresponding results.

First of all, the image thus obtained is passed to morphological operators. These operators play a vital role in object detection as they enhance the image to a greater extent. The morphological operator used for enhancing the image in this system is 'Dilation'. Dilation mask makes the white area in the image to grow, or dilate. Dilation allows the black pixel to remain black only and only if all of the neighbors are black. This operator is used in order to remove the isolated black pixels.

After performing the Dilation operation the enhanced 2-D image is passed to a mathematical operator block known as 'Mean'. This block is used to calculate the mean value of the whole 2-D matrix obtained. This Mean value is the major key to Determine gestures. The Mean value thus obtained is compared with the mean value of the images that are stored in the database.

The comparison of the mean values is done using the 'If' block. This 'If' block takes the mean input value and compares with the range of mean values. If the corresponding mean falls in the range of any particular gesture then corresponding gesture is represented using an image.

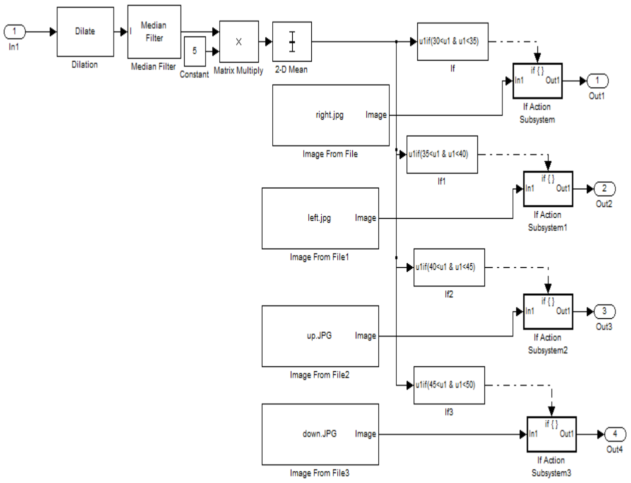
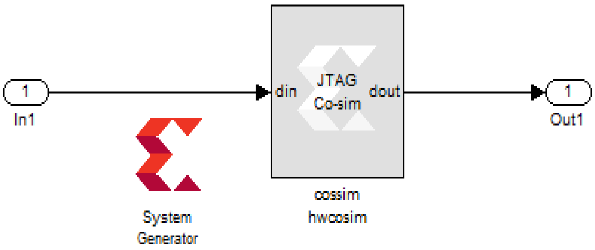


Figure-5. Gesture recognition system.

F. Hardware implementation

After obtaining the required software simulation results, the design can be implemented on Virtex-5 (XC5VLX50T) board. The system generator token can be configured according to our requirements. Within System generator token, Timing and Power Analysis along with HDL Netlist can be analyzed. The algorithm is implemented on FPGA board through JTAG.



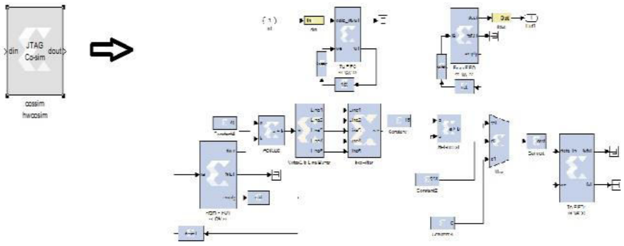


Figure-6. Hardware realization.

RESULTS

The following results were obtained after performing timing and power analysis:

i. Device information

Table-1. Device information.

Family	Virtex-5
Part	xc5vlx50t
Package	ff1136
Temp Grade	Commercial
Process	Typical
Speed Grade	-1

The Device used for performing gesture recognition is mentioned above. The FPGA board used is Genesys Virtex-5 (xc5vlx50t) and the package no. is ff1136 along with the speed grade of -1.

ii. Device environment

Table-2. Device environment.

Ambient temperature (°C)	50.0
Airlfow	250
Heat Sink	Medium Profile
Board Selection	Medium (10"x10")
No. of board layers	8 to 11

The above mentioned data provides the information about the temperature statistics of the FPGA board.

iii. Device utilization

Table-3. Device utilization.

Slice logic utilization	Used	Available	Utilization
No. of Slice Registers	925	28,800	3%
No. of Slice LUTs	579	28,800	2%
No. of occupied slices	356	7200	4%
No. of fully used LUT-Flip Flop pairs	442	1062	41%
No. of bonded IOBs	33	480	6%
No. of Block RAM/ FIFO	12	60	20%
No. of DSP48Es	5	48	10%

The above table provides the utilization of device. For the Virtex-5 (xc5vlx50t-1ff1136) (ML505) board used the no. of slice registers available are 28,800 and out of which only 925 are used which gives a utilization of about 3%. No. of slice LUTs that are used are 579 out of 28,800 and utilization is about 2%. Since our design includes a lot of flip flop pairs it results in utilization of 41% i.e. 442 pairs are used out of 1062. Utilization of block RAM/FIFO is 12 out of 60 which 20% of available Block RAM.

Table-4. Power analysis.

On chip	Power (W)	Used	Available	Utilization
Clock	0.027	1	-	-
Logic	0.003	579	28,800	2%
Signals	0.005	1296	-	-
BRAMs	0.016	12	60	20%
DSPs	0.000	5	48	10%
IOs	0.016	33	480	6%
Leakage	0.448			
Total	0.515			

The above provided table give the complete power analysis of the system. It tells about the power consumed by each and every system that are working in the FPGA board like clock, logic system, BRAMs etc. The total power consumed is equal to 0.515 Watts. If we remove the leakage current the total power consumed will be 0.067 Watts.

Table-5. Supply summary.

Source	Voltage	Total current (A)	Dynamic current (A)	Quiescent current (A)
Vccin	1.000	0.339	0.050	0.288
Vccaux	2.500	0.062	0.000	0.062
Vcco25	2.500	0.008	0.006	0.002

The above mentioned table tells about the input to the system. It tells about the various voltage supply parameters along with the current.

vi. Supply power

Table-6. Supply power.

Total	Dynamic	Quiescent
0.515	0.066	0.448

The total power supplied to the FPGA system is 0.515 Watts out of which 0.066 Watts is Dynamic Power and 0.448 Watts is quiescent power.

vii. Thermal properties

Table-7. Thermal properties

Effective TJA (℃/W)	Maximum ambient (℃)	Junction temperature (°C)
1.5	84.2	50.8

viii. Statistics

Table-8.

Minimum Period	5.570ns
Maximum Frequency	179.533 MHz

SUMMARY

We proposed a gesture recognition system in which a ‘Kinect image’ is passed into the FPGA board by converting the image into serial data. This data is then processed using Xilinx block sets and the desired task of edge detection and thresholding is successfully achieved. The serially processed data is then post-processed in order to regain the image and to get the desired gesture. All these tasks are performed using Xilinx System Generator (XSG) with MATLB and Simulink. It provides rapid means to do hardware implementation of complex techniques used for processing images with minimum resource and minimum delay. Xilinx System Generator (XSG) provides user friendly graphic interface, although the schematic entry is also a GOI interface, Simulink is easier to organize input data and much convenient to observe output in many ways. The RTL schematic of the Xilinx block set is found using ISE 14.3 via Xilinx System Generator (XSG). The Timing and Power analysis report has been validated using ISE 14.3. The Device utilization Summary has also been validated by comparing the values obtained from resource estimator. The hardware is successfully realized on Virtex-5 (xc5vlx50t-1ff1136) (ML505).

The designed architecture used in this project can be used in Real Time Gesture Based Recognition Systems with proper modules in all Xilinx FPGA kit, with proper user configuration in Xilinx System Generator blocks.

REFERENCES

[1] Ms. P.H.Pawar, Prof. R.P. Patil. 2014. FPGA Implementation of Canny Edge Detection Algorithm. Presented in International Journal of Engineering and Computer Science ISSN: 2319-7242, 3(10).

[2] G.T. Shrivakashan, Dr. C. Chandrasekar. 2012. A comparison of Various Edge Detection Techniques used in Image Processing. Presented in IJCSI International Journal of Computer Science Issues. 9(Issue 5, No 1), ISSN (online):1694-0814.

[3] Brian C. Richards, Chen Chang, Robert W. Brodersen work on Programming Streaming FPGA Applications using Block Diagrams in Simulink. from University of California- Berkeley.

[4] Praveen Vanaparthy, Sahitya.G, Krishna Sree. 2013. FPGA implementation of image enhancement algorithms for Biomedical Image Processing. Presented in International Journal of Advanced Research in Electrical, Electronics and Instrumentation Engineering. 2 (11), ISSN (online): 2278-8875.

[5] A.C.Suthar, Mohammed Vayada, C.B.Patel, G.R.Kulkarni. 2012. Hardware Software co-simulation for Image Processing Applications. Presented in IJCSI International Journal of Computer Science Issues. 9(Issue 2, No 2), ISSN (Online): 1694-0814.

[6] Kavitkar S. G., Paikrao P. L. Hardware Implementation of Edge Detection Algorithm-A Review. Presented in International Journal of Engineering Sciences and Research technology” ISSN: 2277-9655.

[7] Mohammed Yousuf Khan. Masarath Nayeem Tayyaba, M.A.Raheem, Ayesha Siddiqua, Syed Sameena. 2014. Image Enhancement and Hardware Implementation of Edge Detected Vascular Images Using Simulink Model. Presented in International Journal of Advanced Research in Computer and Communication Engineering. 3(4), ISSN (online):2278-1021.

[8] S.K.Areefabegam, T.Narendrakumar. 2014. FPGA Based Design and Implementation of Image Edge Detection Using Xilinx System Generator presented in International Journal of New Trends in Electronics and Communication (IJNTEC-ISSN: 2347-7334), 2(8).

[9] R. Gonzalaz, R. Woods. 2002. Digital Image Processing. New Jersey: Prentice-Hall.

[10] DSP System Generator User guide release 12.1. .

[11] Xilinx System Generator User's Guide.

[12] www.xilinx.com.

[13] www.mathworks.com.

[14] www.digilentinc.com/Products/Detail.cfm?NavPath=2,400,819&Prod=GENESYS.

[15] http://www.xilinx.com/support/documentation/sw_manuals/xilinx2013_1/ug948-vivado-sysgen-tutorial.pdf.