DESIGN AND IMPLEMENTATION OF EMBEDDED VISION BASED TRACKING SYSTEM FOR MULTIPLE OBJECTS USING FPGA-SOC

Saif N. Ismail[1], Muataz H. Salih[1] and Wahab Y.[2]
[1]School of Computer and Communication Engineering, University Malaysia Perlis Universiti Malaysia Perlis (UniMAP) Perlis, Malaysia
[2]Adv. Multidisciplinary MEMS Based Integrated Electronics NCER Centre of Excellence, School of Microelectronic Engineering, University Malaysia Perlis Universiti Malaysia Perlis (UniMAP) Perlis, Malaysia
E-Mail: sai91f@gmail.com

ABSTRACT

An implemented vision based tracking system has become an important application of embedded in the field of vision and streets surveillance systems in the field of public security. That is why many of researchers have been suggested different embedded vision tracking approaches. This project addressing a significant issue namely multiple objects tracking. One of significant problems that are faced the researchers is multiple objects tracking which is addressed in this project. Therefore, this project is designing and implementing an embedded vision based for multiple object (color) tracking system using FPGA-SoC. The proposed method has adopted a passive tracking vision system based on platform DE1-SoC and D5M camera. As a result of our project is can be tracking distance of multiple objects (colors) was reached up to 30 meter for sized 15×15 cm object shape.

Keywords: embedded vision, FPGA system tracking, multi-object.

INTRODUCTION

Embedded systems can be easily find everywhere in our daily lives, have become very important as they are present in almost every aspect of modern life; telecommunication systems, computer networks, consumer electronics, household appliances, heating control systems, traffic lights, engine control systems, industrial automation controllers, medical devices etc. all make use of embedded systems to provide flexibility, efficiency and complex features. According to the World Semiconductor Trade Statistics today about 98 percent of programmable digital devices are actually embedded.

An embedded system is a PC framework with a devoted capacity inside a bigger mechanical or electrical framework, frequently with ongoing figuring imperatives [1]. It is installed as a feature of a total gadget regularly including equipment and mechanical parts. Inserted frameworks control numerous gadgets in like manner utilize today. Ninety-eight percent of all chip are manufactured as components of embedded systems [2].

Cases of properties of commonly inserted embedded PCs when contrasted and universally useful partners are low power utilization, little size, tough working extents, and low per-unit cost. This comes at the cost of constrained preparing assets, which make them altogether harder to program and to communicate with. Notwithstanding, by building knowledge components over the equipment, exploiting conceivable existing sensors and the presence of a system of implanted units, one can both ideally oversee accessible assets at the unit and system levels and also give expanded capacities, well past those accessible. For instance, clever procedures can be intended to oversee control manage power consumption of embedded systems [3].

As well as a characteristic of an embedded system unlike general purpose computing systems, [4] embedded systems possess in specific characteristics and these characteristics are unique to each embedded system.

Active Type Tracking System Active type tracking system is one kind of tracking which is able to send out the energy and measure the return values. Corresponding action will be taken according to the magnitude of the measurement. In [5], some researchers surveyed about single and multi-tracking methodologies that utilized distance-only, bearing-only, and both distance and bearing measurements.

RELATED WORK

FPGA is a logic chip that can be programmed and contains of thousands of logic gates depicted in Figure-2.3. It is an FPD featuring a general structure that allows higher logic capacity than any other programmable logic devices. The FPD stated here is a short form of Field-Programmable Device which is referring to any type of integrated circuit utilized for implementing digital hardware. FPD can be also known as Programmable Logic Devices (PLDs). Programmable Logic Array (PLA) is a moderately tiny part of FPD that contains two main levels among all of the logics, AND-plane and an OR-plane

The Programmable Logic Devices can be classified into three categories:

a) Simple Programmable Logic Devices (SPLDs)
b) Complex Programmable Logic Devices (CPLDs)
c) High-Capacity Programmable Logic Devices (HCPLDs)

The emerging of Mask Programmable Gate Arrays (MPGAs) has given impetus to the motivation user-programmable equivalent design, which is FPGAs. MPGA was introduced to overcome the problem of handling the circuits which are having a huge logic. It is a traditional version of FPGA which consists of an array of prefabricated transistors. By connecting custom wires to the relevant transistors, it is able to be reformed into user's logic circuit. The most significant difference of MPGA compared to FPGA, is the customization of MPGA can only be performed during the process of chip fabrication and can be only done on the manufacturing plant side, while FPGA enables any type of users make the changes of the logic structures as the chip can be reprogrammed all over and over again as long as the chip remains functionally, with the exception of those FPGAs which are anti-fusing. FPGAs constitute a variety of uncommitted circuit components, for instance, logic blocks as well as interconnected assets. Present day FPGAs comprise of blends of configurable Static Random Access Memory (SRAM or Flash), logic blocks, high-speed input/output pins (I/Os), as well as routing. Figure as below illustrated an emblematic FPGA's architecture. As the main sort of Field Programmable Devices (FPDs) that is backings a very high logic limitation, FPGAs have been culpable for major shifting as the way of computerized circuits are outlined Figure-1.

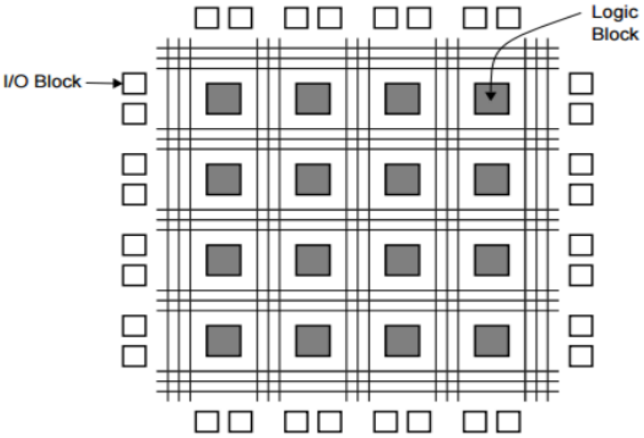


Figure-1. Structure of FPGA [6].

Through contrasting FPGAs to other different methods for constructing typical hardware, FPGAs have two interesting points. First, they empower users to build precisely the hardware required by users, rather than proceeding with the obsolete method, Application-Specific Standard Product (ASSP) which is already widely used by many contenders, or needing to embrace the cost, time and risk of an Application-Specific Integrated Circuit (ASIC) design. Second, FPGA"s design flow takes out the complexity and tedious floor plane extracting, place and course, timing analysis, and masked phases of the undertaking since the logic designed is already integrated to be put onto an effectively checked, characterized FPGA device. Interpret the design flow at Figure-2 showed comparison between FPGA and ASIC. In conclusion, FPGAs manage to handle up operation quickly, simply and energy-efficiently.

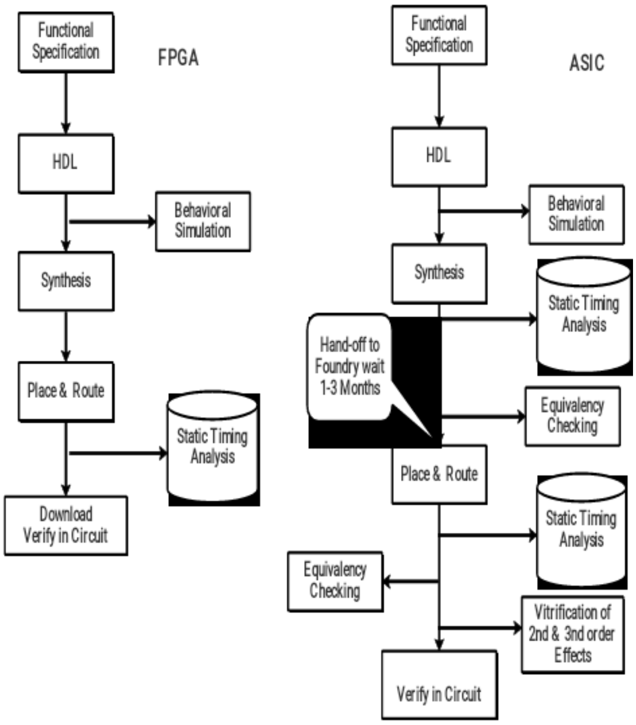


Figure-2. Design flow comparison between FPGA and ASIC [7].

The researchers were trying to construct a Tracking system is used for the observing of persons or objects on the move and supplying a timely ordered sequence of respective location data to a model e.g. capable to serve for depicting the motion on a display capability. Tracking of objects from images is a tough task. Therefore stereo vision system is incorporated in proposed design so this stereo vision system helps to identify the states of object which will be helpful for tracking of objects. As tracking needs motion of the object, that part is also being incorporated in proposed design. For increasing the speed of the design efficient hardware architecture will be design using HDL. They suggested the image processing can improve the efficiency of tracking system design and implementation of a visual servo system, which uses a stereo machine vision system to guide an omnidirectional mobile robot for real-time object tracking. However, the computational complexity and large amount of data access make real-time processing of stereo vision challenging because of the inherent instruction cycle delay within conventional computers. The entire stereo vision process, for example, amendment, stereo coordinating, and post-preparing, is acknowledged utilizing a solitary field programmable entryway exhibit (FPGA) without the need of any outer gadgets. Stereo vision has generally been and keeps on being a standout amongst the most broadly explored themes in PC vision. Since stereo can give profundity data, it has potential uses in numerous visual spaces, for example, self-governing route, 3D recreation, protest acknowledgment, and observation frameworks. Particularly, it is likely the most broadly utilized for robot route in which precise 3D data is urgent for the unwavering quality of route. In any case, FPGAs have just demonstrated their superior limit with respect to picture preparing errands particularly for inserted frameworks. This makes the FPGAs offer great flexibility in manipulating [8].

DE1-SoC VS. OTHER PLATFORM

In related works overview, we considered numerous papers. The various technique to object tracking.

A. De1-SoC vs. Raspberry Pi

Researchers [9] had work on focus object tracking with partial or full occlusion. Object tracking is finished utilizing highlights like hues and forms. We have proposed a strong Color-based calculation to track the question and handle impediment continuously space. In this algorithm, HSV color model is utilized and HSV scope of the shading item to be followed is chosen. Required question territory is recognized and shape is framed appropriately to track the objects. Proposed calculation is actualized on ARM Cortex-A7 utilizing Open Source Computer Vision (OpenCV) and Linux-embedded platform. The investigation is finished utilizing produced genuine database and unusual crowd activity movement database.

Also the proposed suggestion on [10] algorithm to track the line on the ground in view of PC vision, which uses Raspberry Pi 2 to finish the image processing. The algorithm work on image processing first initially, then defined line by edge identification and filtering model. The proposed algorithm can be utilized as a part of an assortment of platforms included intelligent car system, robot, UAV and other versatile platforms. In this proposed, the algorithm is confirmed by quad rotor stage and accomplishes big line-following action by setting the parameters. Arrangement of analyses demonstrates that the algorithm has major validity, hardiness and generality.

B. De1-SoC vs. Arduino

The system vision embedded is able to track any completely shaded item utilizing the shading based following calculation at an ordinary frame rate of 25 frames per second, which is adequate for real-time application. For the testing of template based tracking method, the robot missed the tracking target sometimes. If the object is moving at a higher speed, the higher the possibility the robot would missed the tracking target.

The best solution to counter this issue is to increase the processing power of the Android device by virtue of the tracking based algorithm will consume more process power than color tracking algorithm.

Object tracking regularly includes the utilization of camera to catch the movement of the items which are to be followed. It is utilized to discover the position of the items in the acquired image signals. Diverse positions of the related items can be delegated distinctive signs to a framework which is utilizing the object tracking.

It is a vital undertaking in the field of computing vision. The accessibility of less expensive cameras and advances in the calculations of image processing has made the utilization of object tracking in a substantial number of genuine applications. From the algorithm of object tracking, object can be anything that is may be enthusiasm for further and future examination. The portrayal of the objects in image can be in wide range of structures. Typically objects are portrayed to by their appearances like shape and color. The position of an object can be spoken to from the point (centroid) or an arrangement of focuses. Principally, the point (centroid) is useful for tracking an object that possesses tiny regions in an image [11].

SUGGESTED METHODOLOGY

The project was conducted according to the planned phases. A good understanding of existing and relevant knowledge was an important primer to commence the project. The entire project flow after planning is illustrated in Figure-3 below.

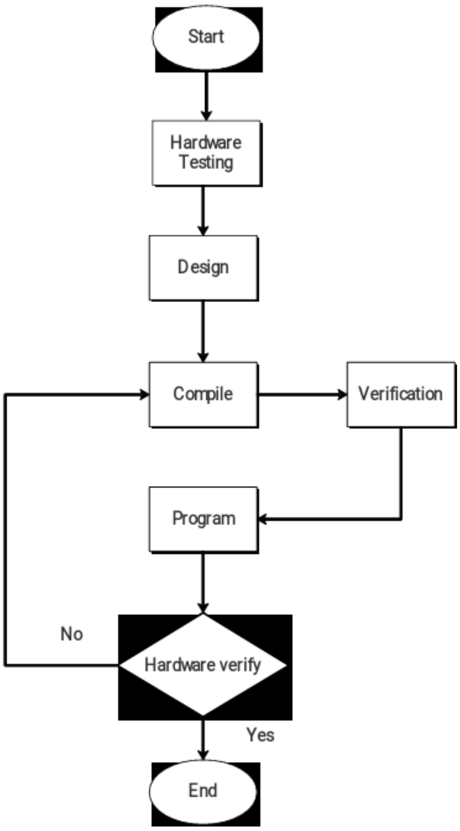


Figure-3. Overall project flow.

The first step was to study the VGA port and to understand its working principles, and then the display and interfacing devices were studied before designing them and testing them. This was important to ensure there were no faults with the hardware before implementing the software. Next came the design steps, in two stages:

a) Interfacing the devices and components.

b) Designing an algorithm for image, video and signal processing.

This project used both Verilog and VHDL code. At this point we loaded the schematic from the Terasic company that works on the D5M camera and the DE1-SoC platform. We then proceed to carry out compilation, simulation, programming and verification of the work as proposed and shown in Figure-4.

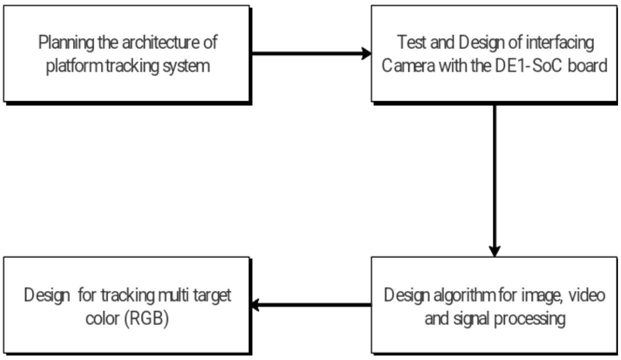

Planning the architecture of platform tracking system
Test and Design of interfacing Camera with the DE1-SoC board
Design algorithm for image, video and signal processing
Design for tracking multi target color (RGB)

Figure-4. Planning to design algorithm.

A. Block diagram of component tracking system

A top-level architecture design block diagram is provided. The signal acquired from the camera sensor is processed via image processing and converted into a digital signal if the signal acquired is analogue. After the processing, the data is sent to the central computing unit of the FPGA controller as an input value to decide the action, motion and for displaying output to the displaying unit (e.g. VGA Monitor). As shown in Figure-5 for the correct operation of the project, one needs:

a) DE1-SoC FPGA board
b) D5M CMOS sensor
c) VGA monitor

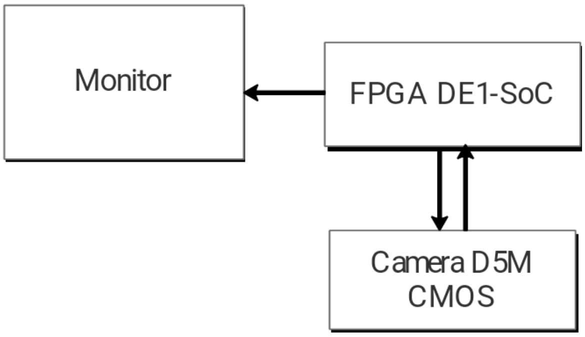


Figure-5. Top-level architecture design, block diagram.

C. D5M Camera testing

Device testing is the first thing to do to ensure that the hardware is working well. Different devices have to undergo individual testing procedures and most of these procedures are found in the manual or handbook of the devices Figure-3.5.

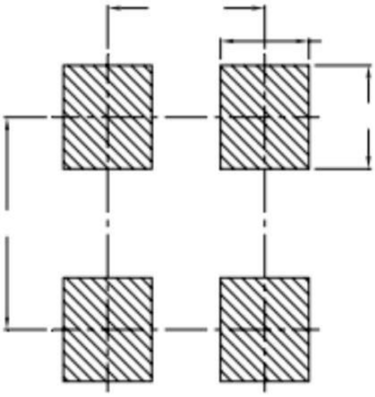


Figure-6. Recommended land [12].

The camera used as the main sensor for tracking purposes is the 5-megapixel camera package from Terasic. It is equipped with a Micron D5M CMOS sensor and supports 2592H x 1994V active pixels. The captured output is in RGB Bayer Pattern format and has a full resolution, a frame rate of up to 15 frames per second (FPS). The VGA monitor VGA (640 x 480), with binning display resolution, provides the output viewing function. The D5M Camera Package and VGA monitor module are shown in Figure-6. The Monitor VGA module is mainly controlled by a parallel data flow, which is in RGB digital intensity. It is synchronised at the frame level with vertical synchronisation and synchronised at the line level with horizontal synchronisation all synchronised by a DOT CLK clock at the pixel level.

Figure-7. D5M GPIO Camera.

For interfacing the devices, testing was undertaken by implementing some simple programming. After interfacing in Figure-7 and the monitor, VGA and camera module along with the main board DE1-SoC had to be prepared in order to design a test for them, as shown in Figure-8.

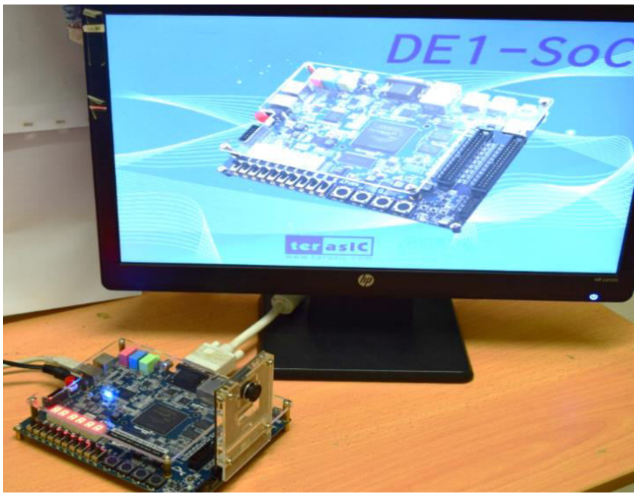


Figure-8. FPGA DE1-SoC platform interfaced with the camera module.

The algorithm was designed using the Verilog programming language and run on the Cyclone V processor. Terasic has custom made this to display images on the VGA monitor. The actual image data is transferred from the D5M camera to the DE1-SoC board. Then the DE1-SoC board carries out the image processing and converts the acquired data into the RGB format in order to display the image on the VGA monitor. The testing, demonstration setup has been found in the D5M camera manual.

D. Algorithm for image and data processing

There are two major components of a visual tracking system, distance and object size, as well as filtering and data association. The proposed algorithm for object tracking in this project is the Particle filter. The solution search space is searched by using particles captured from the images, in other words, it is doing sampling by using the particles. Depicts the two main steps of the Particle Filter, predict and correct at Figure-9.

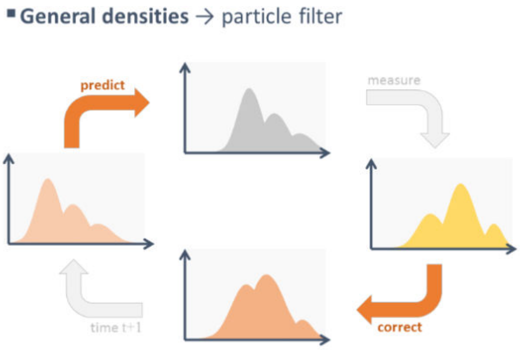


Figure-9. Two main steps of particle [13].

D5M CMOS Camera interfacing

This section explains the components that interface the camera with the board, to send the captured image data from the sensor to the DE1-SoC board and transform it from Bayer colour format to RGB format before storing the data in the SDRAM of the DE1-SoC. Figure-10 shown below depicts the two parts of the system, the D5M CMOS image sensor and the camera controller.

The image data is first captured by the camera's CMOS sensor and is forwarded to the image capture module, where the valid pixel information is extracted according to the validity control signals.

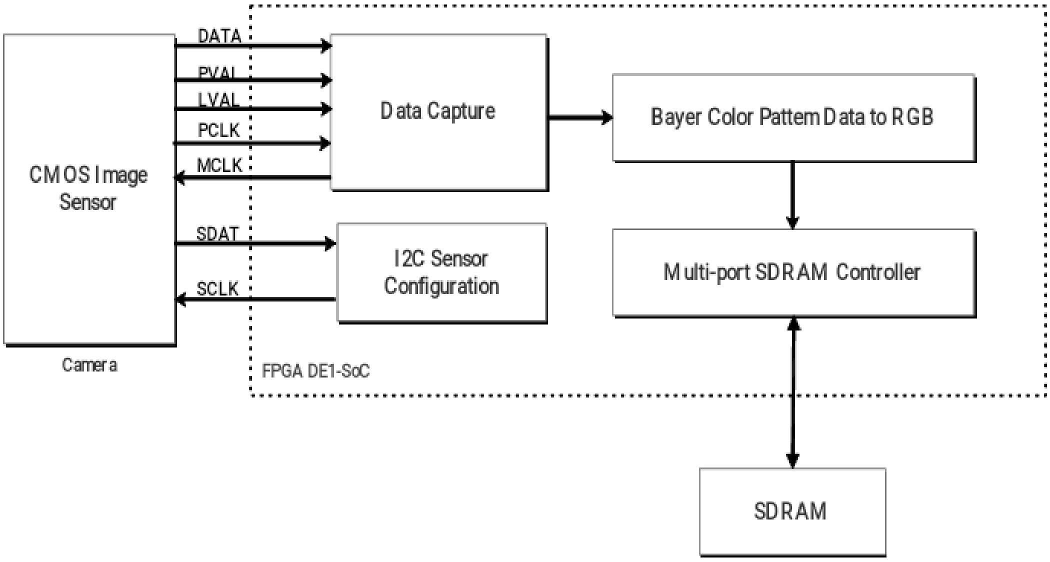


Figure-10. The camera interface block diagram.

E. SDRAM Controller

Also, attached to the FPGA fabric is an external 64MB SDRAM (32Mx16) memory chip. This memory chip is connected to the FPGA via a DRAM_DQ 16-bit data bus, control lines, and a DRAM_ADDR 13-bit address bus. The FPGA to SDRAM interface is displayed in Figure-11. The timing and control signals for memory chips are significantly more complicated than those for a

VGA connection. Therefore, this project uses a pre-made SDRAM controller module from the DE1-SoC example code which is responsible for connecting the data to the much simpler interface exposed by the SDRAM controller.

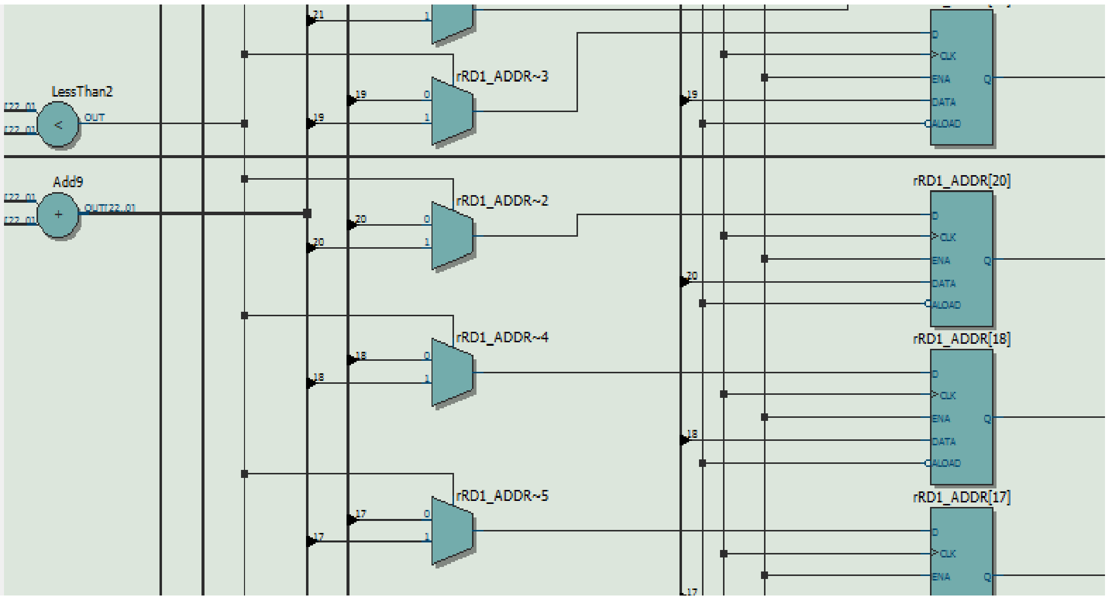


Figure-11. RTL View design in SDRAM.

The Tables below contain the main configurations for all the 4 FIFOs. For simplicity, we will call them FIFOs 1 to 2 and have a write and a read version of each. We will now explain some of the values used in Table-1.

Table-1. The main configurations FIFOs 1 to 2.

	FIFOs1		*FIFOs2*	
	Wirte	*Read*	*Write*	*Read*
Data	*sCCD_G[11:2],sCCD_R[11:2]*		*sCCD_G[11:7],sCCD_B[11:2]*	
Max Address	*640*480*		*22'h100000+640*480*	
Clock	*CCD_PIXCLK*	*VGA_CTRL_CLK*	*CCD_PIXCLK*	*VGA_CTRL_CLK*

Table-2. The main configurations FIFOs 3 to 4.

	FIFOs3		*FIFOs4*	
	Wirte	*Read*	*Write*	*Read*
Data	*Frame: Read_DATA1*		*Frame: Read_DATA2*	
Max Address	*640*480*		*22'h100000+640*480*	
Clock	*CCD_PIXCLK*	*VGA_CTRL_CLK*	*CCD_PIXCLK*	*VGA_CTRL_CLK*

FIFOs1 and FIFOs2 are responsible for storing and retrieving the current frame, using the SDRAM as frame buffer, and FIFOs3 and FIFOs4 are responsible to save and retrieve the background. Each FIFOs store half of the information of a pixel and thus the reading must be concurrent as such show in Table-2.

The clock used to read from the SDRAM is the negation (not) of the VGA_CTRL_CLK signal, which is the clock used to control the VGA pixel clock. For writing, the CCD capture pixel clock was used.

The starting addresses correspond to the first address of each bank, to which have added 640x480 the Max Address - to store the amount of pixels needed for the VGA display.

F. Image storage

The image's RGB data is written to the SDRAM with the help of the four buffering FIFOs and the SDRAM controller. The use of the FIFOs for reading and writing is necessary to overcome clock mismatches between the camera's pixel clock, the SDRAM and the VGA clock. For storage purposes, each pixel's data is segmented into two halves one containing all of the red and a half of the green pixel information, and the other containing all of the blue and the remaining half of the green pixel information. Each half is buffered to an individual FIFO, and each FIFO writes to a designated SDRAM bank. A pair of writing FIFOs holds the pixel data regarding the current image and the other pair holds the image data of the background. The same goes for the four reading FIFOs. The reading FIFO stores the RGB data concerning the current and background frames read from the SDRAM as shown in Figure-12.

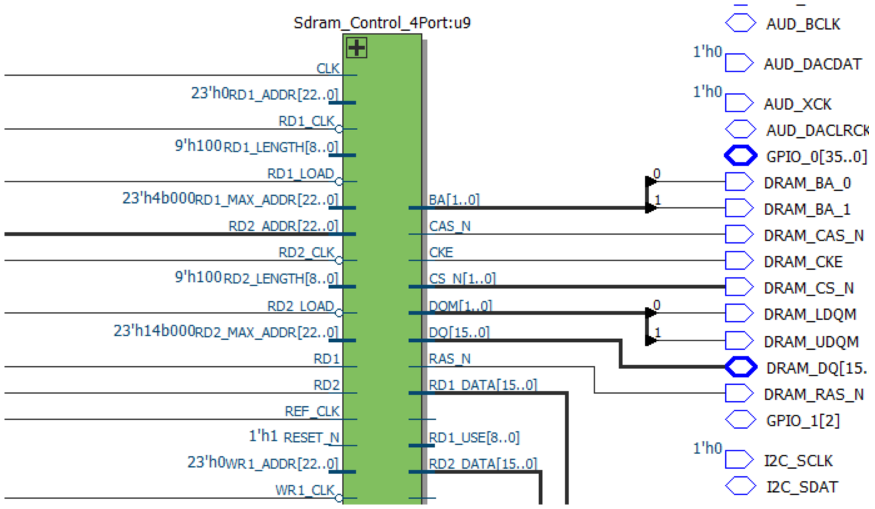


Figure-12. RTL View design image storage.

G. Image capture

The image data is first captured by the camera's CMOS and is forwarded to the image capture module, where valid pixel information is extracted according to the validity control signals. The resulting RAW data is then converted to RGB format. The view of the design image capture of RTL is shown in Figure-13.

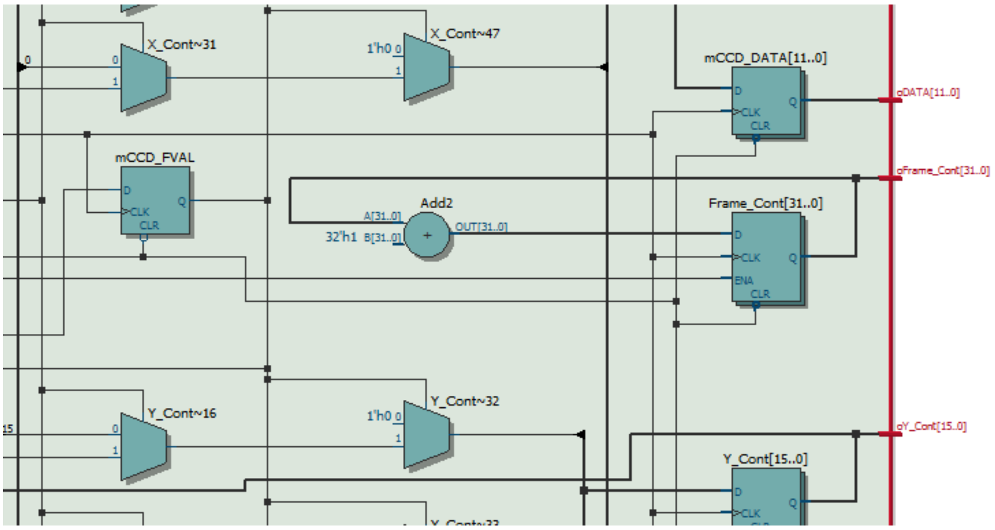


Figure-13. RTL View design image capture.

H. VGA Controller

In this part to explain have been discussed in chapters 2, the platform DE1-SoC includes a connection VGA which have 15 pin DC represent output with a component that has synchronous signals are provided directly from in chip Cyclone V devices AD7123 using threefold 10 bit high speed to generate analogue data signals are (red, green, blue) gives the associated layout shown in Figure-14.

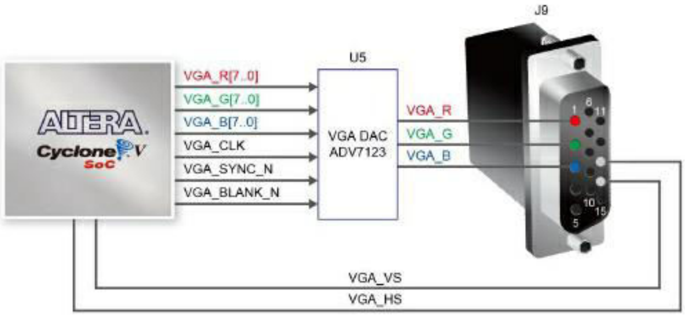


Figure-14. VGA Display interface between FPGA and display connector [14].

In summary, from the output of the VGA controller, we obtained the current frame and the results of the subtraction of the current frame with the background in RGB. The RTL simulation is shown in Figure-15.

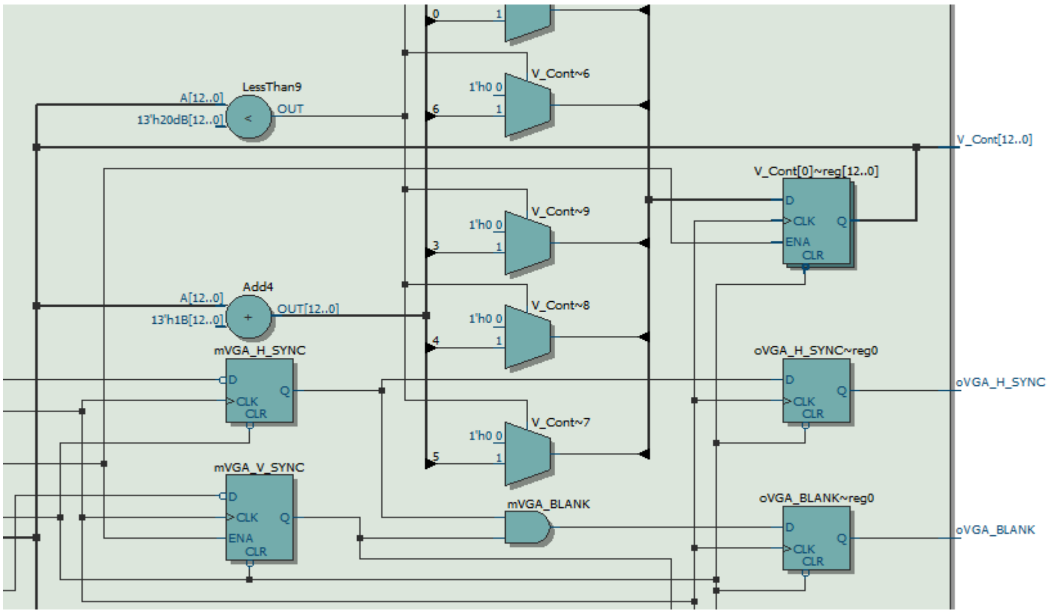


Figure-15. RTL View design VGA control.

I. Image post-processing

After reading the image's RGB data from the SDRAM, the absolute difference in value between the reference background and the current frame for each pixel is calculated in the VGA controller and is then output to a thresholding operation which binarizes the pixel value as shown in Figure-16.

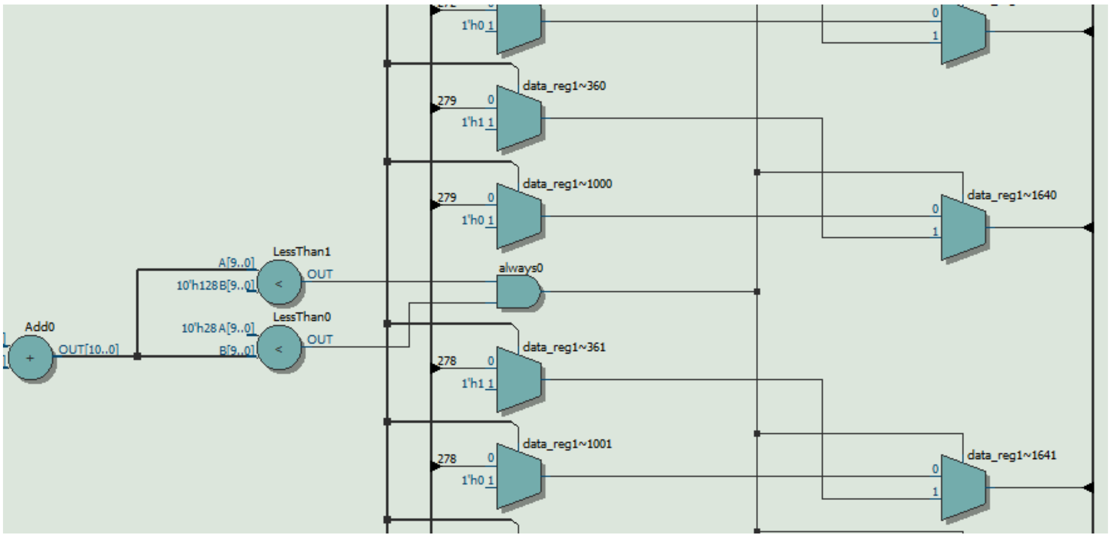


Figure-16. RTL View design image post-processing.

J. Image display

This section explains the displayed data which depends on the selected threshold value for binarisation and on the selected colour mode. If the selected colour mode is the binarised difference image, the binarised image and red contours obtained in the described image post-processing phase have been displayed. Otherwise, the original RGB object data and the same red contours have been displayed as shown in Figure-17.

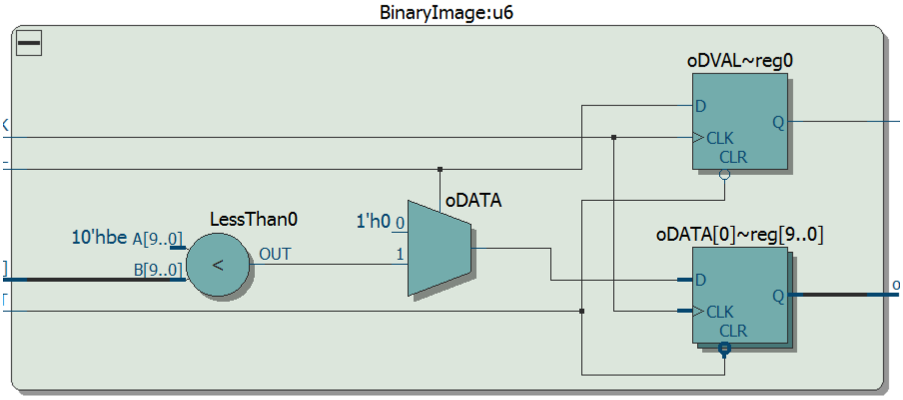


Figure-17. RTL View design image display.

K. Grayscale unit

In this section, the image is processed and displayed on the screen after being processed in the previous sections; the resulting image requires some additional processing in order to reduce the noise and to highlight the object. This was started from the part module image storage to output by converting the image differences into a greyscale, over which a simple threshold was applied to reduce false positives and noise, the conversion implemented is a simple average of the three RGB components as shown in Figure-18. The conversion to the greyscale image and the application of the colour control equation were recommended.

$$Gray\ level = \frac{(R+G+B)}{3} \qquad (1)$$

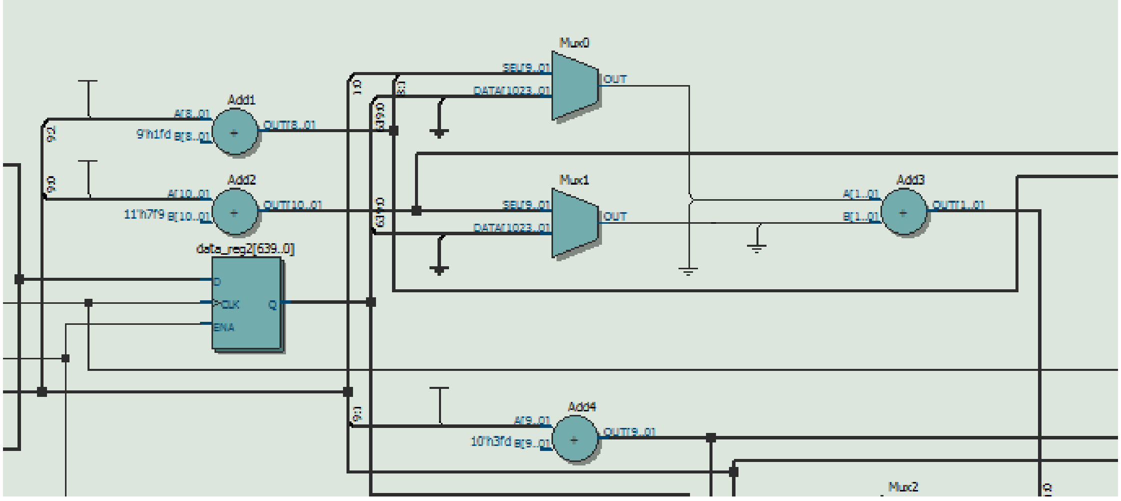


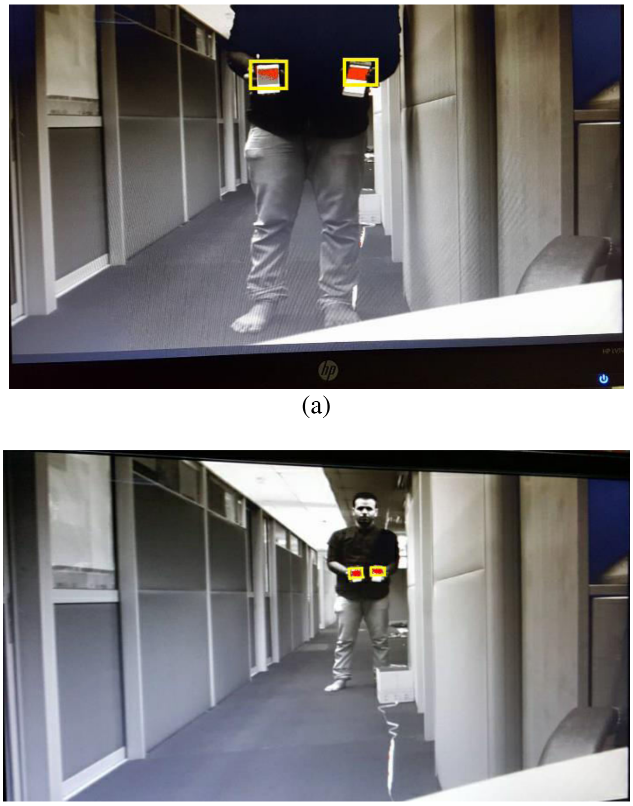
Figure-18. RTL View design gray unit.

RESULT

There are two color have been tested

A. Red color experiment

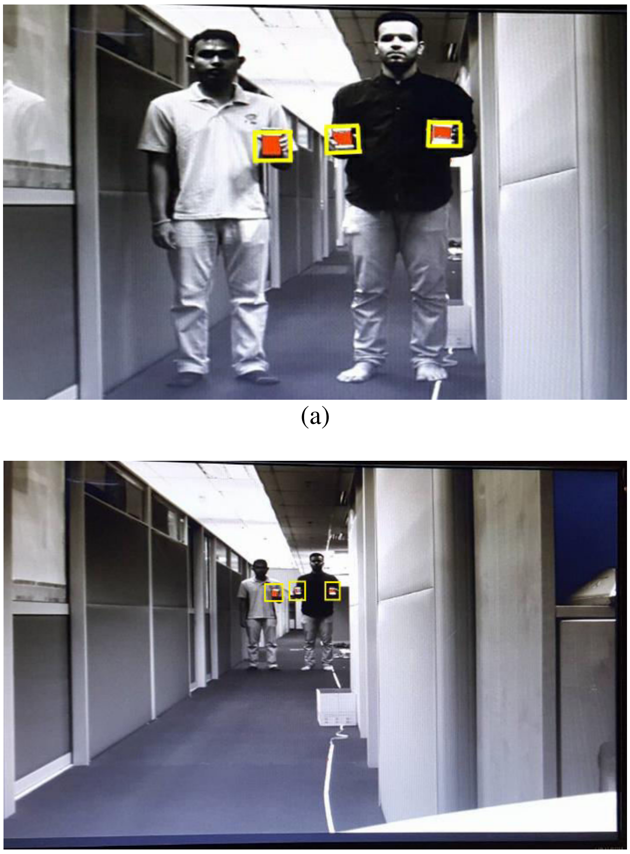
The tests here were carried out on red colour, the 10 tests start at a minimum of 15 cm to a maximum of 30 m. In Figure-19 the results in the yellow box represent the tracking of two red coloured object at a distance of 2.5 m, (b) shows the tracking of two red coloured objects at a distance of 6 m.


(a)

(b)

Figure-19. Tracking two red colour objects at a distance of (a) 2.5 m and (b) 6 m.

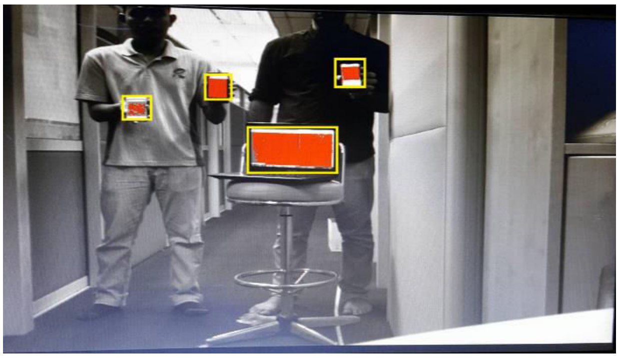
In this excrement at the Figure-20 represent the yellow boxes show three red colour objects tracked at same time at a distance of (a) 4 m and (b) 10 m.


(a)

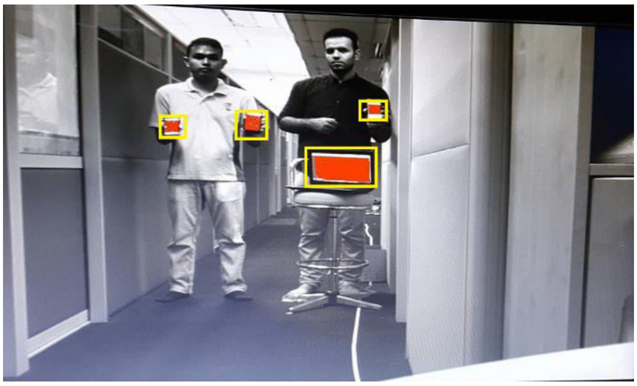
(b)

Figure-20. Tracking three red colour objects at a distance of (a) 4 m and (b) 10 m.

In the Figure-21. The yellow boxes denote tracking of four red objects at the same time at a distance of (a) 2.5 m and (b) 4 m.



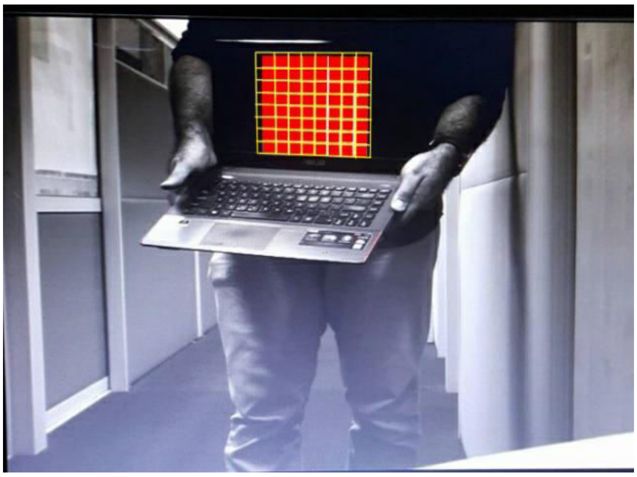
(a)



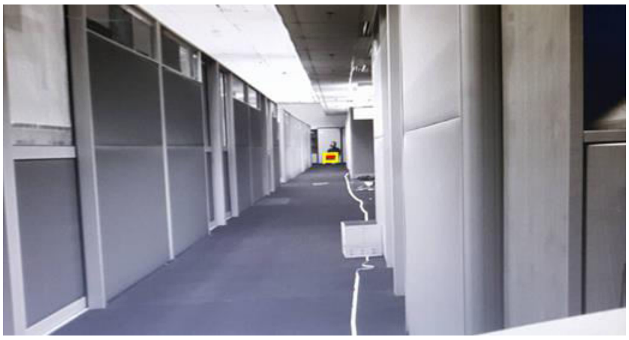
(b)

Figure-21. Tracking four red colour objects at a distance of (a) 2.5 m and (b) 4 m.

In the Figure-22. Shows the tracking of multiple red objects at the same time (a) at a distance of 2.5 m (b)at the maximum test distance of 30 m. The object was sized at 30×15 cm.



(a)

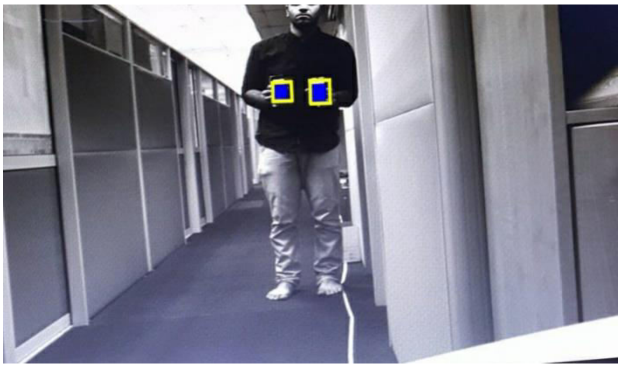


(b)

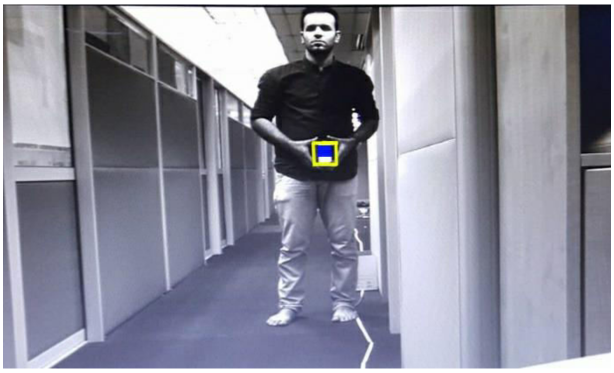
Figure-22. Tracking multiple red objects at a distance of (a) 2.5 m, (b) 30 m.

B. Blue color experiment

In Figure-23 the results in the yellow box represent the tracking of two red coloured object at a distance of 2.5 m, (b) shows the tracking of two red coloured objects at a distance of 6 m.



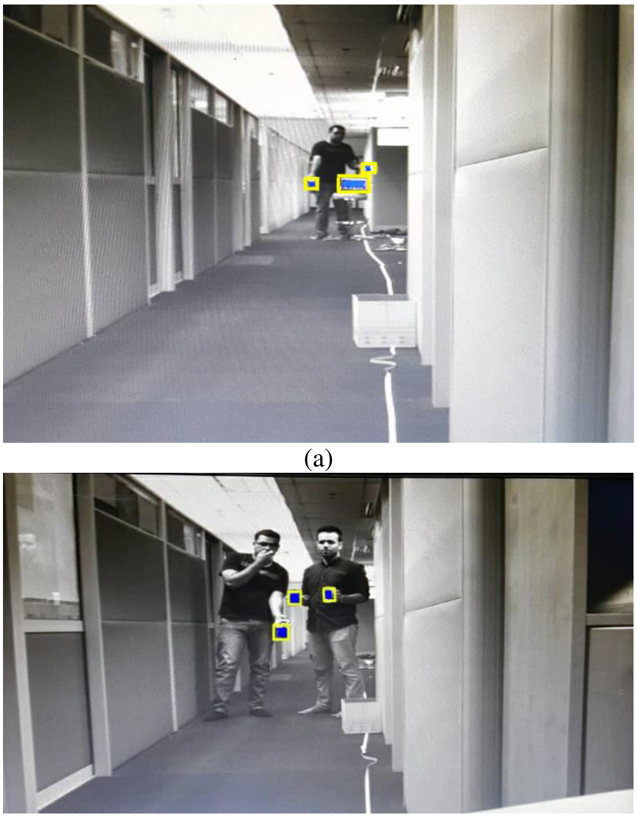
(a)



(b)

Figure-23. At a distance of 4 m tracking (a) two blue objects, (b) one blue object.

In this test at a distance of (a) 15 m and (b) 6 m as shown in Figure-24 the yellow boxes denote that the blue colour tracking has tracked three objects.

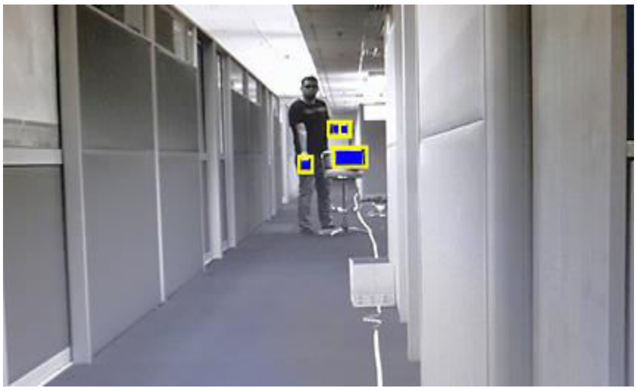


(a)

(b)

Figure-24. At a distance of (a) 15 m and (b) 6 m three blue objects were tracked.

In the Figure-25 at a distance of (a) 15 m and (b) 6 m the experiment showed that four objects were tracked at the same time with the same color.

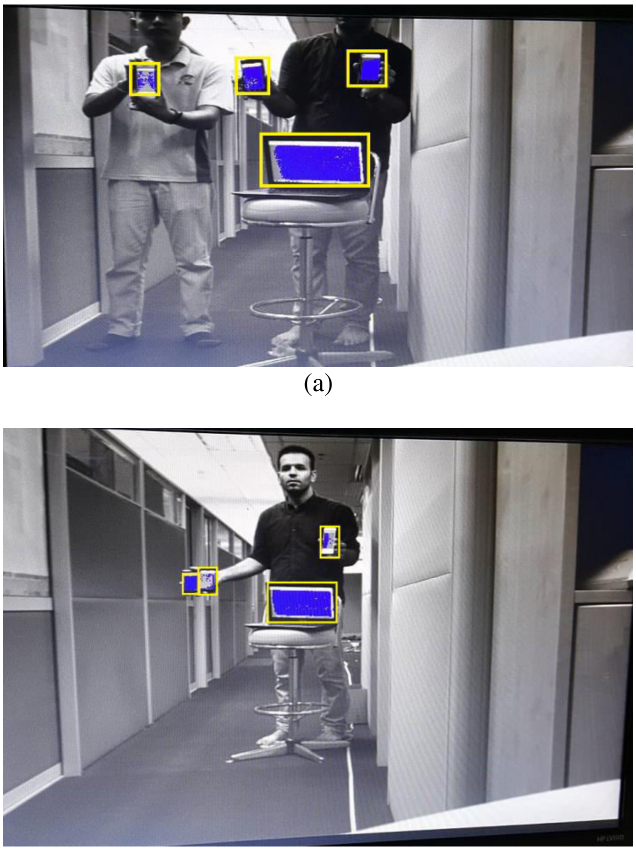


(a)

(b)

Figure-25. At a distance of (a) 10 m and (b) 6 m four blue objects were tracked.

In the Figure-26, the four blue objects shown in yellow boxes were tracked at a distance of (a) 2.5 m and (b) 4 m.

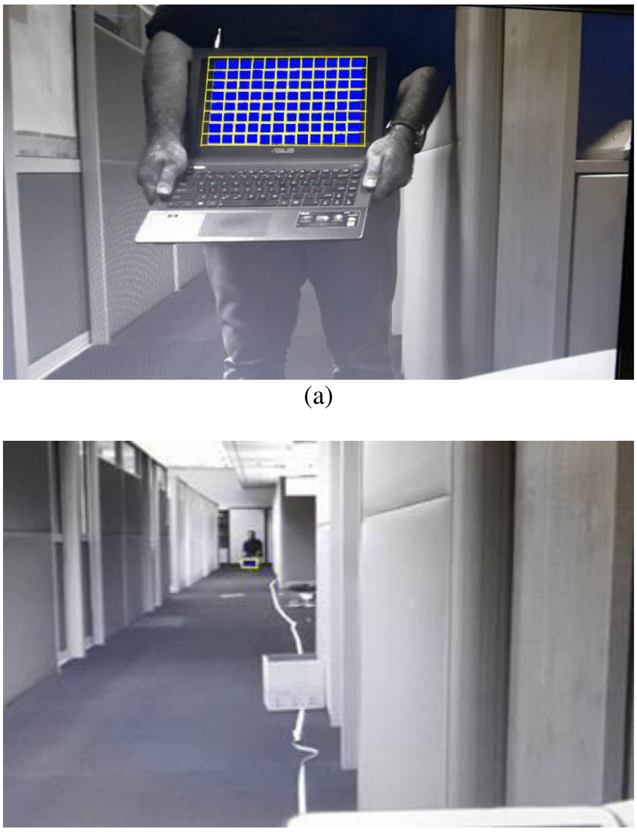


(a)

(b)

Figure-26. At a distance of (a) 2.5 m and (b) 4 m four blue colour objects were tracked.

Another experiment is shown in Figure-27(a) we see multiple blue objects were tracked at the same time at a distance of 2.5 m and (b) at the maximum test distance of 30 m. The object was a sized at 30×15 cm.


(a)

(b)

Figure-27. (a) Multiple blue objects tracked at a distance of (a) 2.5 m and (b) at a distance of 30 m.

CONCLUSIONS

In our project presented tracking system based on FPGA in real-time, implement a passive embedded based vision tracking system based on FPGA-SoC and verify and optimize images. The proposed of method is consist platform DE1-SoC and camera D5m have been implemented, in the idea of this project is making every frame grayscale, color and the color definition which would be tracked and the image processing filtering particle have been applied. the result of our this project can be tracking a multiple object and multiple color in addition to the distance of this project that reach 30 meters size of the object has used 15×15 cm. finally as well as the platform of the DE1-SoC work on highly frequency 1.6 Ghz.

ACKNOWLEDGEMENT

The authors would like to thank the Ministry of Education Malaysia (MOE) for providing the FRGS research grant (Grant no. 9003-00474).

REFERENCES

[1] Bharani M., Elango S., Ramesh S. & Preetilatha R. 2014. An Embedded System Based Monitoring System For Industries By Interfacing Sensors With ATmega Microcontroller. International Journal of Advanced Research in Electronics and Communication Engineering (IJARECE). 3: 1472-1474.

[2] SHRUTI V. 2016. Recent Trends in Embedded System in Electronic and Communication. PARIPEX-Indian Journal of Research. 5(7).

[3] Mittal S. 2014. A survey of techniques for improving energy efficiency in embedded computing systems. International Journal of Computer Aided Engineering and Technology. 6(4): 440-459.

[4] Kamal, R. 2008. Characteristics & Quality Attributes of Embedded Systems. 30 -32.

[5] Zhou K. & Roumeliotis S. I. 2011. Multirobot active target tracking with combinations of relative observations. IEEE Transactions on Robotics. 27(4): 678-695.

[6] Ellis III F. E. 2011. Computer or microchip protected from the internet by internal hardware: Google Patents.

[7] Briggs E. 2015. FPGA Digital Music Synthesizer. Worcester Polytechnic Institute.

[8] Ding J., Liu J., Zhou W., Yu H., Wang Y. & Gong X. 2011. Real-time stereo vision system using adaptive weight cost aggregation approach. EURASIP Journal on Image and Video Processing. 2011(1): 20.

[9] Mandal S. 2014. P. E. T - Pi Enabled Tracking Bot.

[10] Yang H., Wang Y. & Gao L. 2016. A general line tracking algorithm based on computer vision. Paper presented at the Control and Decision Conference (CCDC), 2016 Chinese.

[11] Rautaray S. S. & Agrawal A. 2015. Vision based hand gesture recognition for human computer interaction: a survey. Artificial Intelligence Review. 43(1): 1-54.

[12] Mehi J., Daigle R. E., Brasfield L. C., Starkoski B., Wen J., Liu K. W., Hirson D. 2011. High frequency array ultrasound system: Google Patents.

[13] Afsar P., Cortez P. & Santos H. 2015. Automatic visual detection of human behavior: a review from 2000 to 2014. Expert Systems with Applications. 42(20): 6935-6956.

[14] Kashani-Akhavan S. & Beuchat R. SoC-FPGA Design Guide.