IOSR Journal of VLSI and Signal Processing (IOSR-JVSP)
Volume 7, Issue 5, Ver. I (Sep-Oct. 2017), PP 12-22
e-ISSN: 2319 – 4200, p-ISSN No. : 2319 – 4197
www.iosrjournals.org

FPGA Based Moving Object Tracking For Indoor Robot Navigation

T. D. Magdum * [a], P. C. Bhaskar [a]
[a]*Department of Technology (Electronic), Shivaji University, Kolhapur -416 004, India*
Corresponding author: T. D. Magdum

***Abstract**: Indoor environments such as houses, offices, hospitals, mobile robots have to be equipped with a capability to navigate in indoor environments to execute a given task while avoiding obstacles. A number of sensors are used widely in order to navigate while detecting obstacles in indoor environments. However, most of these sensors are too expensive to apply for low-cost service robots. Thus we can use low cost surveillance camera for indoor robot navigation using the visual navigation. This paper gives the state of the art the FPGA and indoor robot navigation concept with the focus on FPGA based moving object tracking. The paper starts with an overview of FPGA base image processing in order to get an idea about FPGA architecture, and followed by an explanation on Moving object tracking algorithm and virtual path calculation. Finally, we concluded FPGA is an ideal choice for implementation of visual navigation for real time moving object tracking algorithms.*

***Keywords:** FPGA implementation, Indoor navigation, Moving object tracking algorithm, Virtual path claculation*

Date of Submission: 28-09-2017 Date of acceptance: 12-10-2017

I. Introduction

Moving object tracking is one of the fundamental components of computer vision; it can be very beneficial in applications such as unmanned aerial vehicle, surveillance, automated traffic control, biomedical image analysis, intelligent robots etc. The problem of object tracking is of considerable interest in the scientific community and it is still an open and active field of research [1] (Shashank Pujari; 2008).

Most of the image processing algorithms are sequential in nature. These algorithms are most suited in applications, which do not have any time restrictions. In other words, where the response time is not so important. In real-time systems, sequential algorithm will not be a good choice due to time and resource constraints.

Field Programmable Gate Arrays (FPGA) on the other hand gives a platform for parallel execution. In an FPGA based design, different hardware blocks execute the sequences of an algorithm in parallel, and thus provide quick response and high frame rate. Since the overall operations are performed in less number of clock cycles, the power consumption will be reduced considerably, compared to micro-controller/DSP-processor based designs.

1.1 Motivations

In a typical micro-controller/DSP processor based design, executes algorithm sequences sequentially. This will involve storing the frames in a buffer, and then performing the operation. If multiple hardware circuits can be designed to carry out different algorithm sequences in parallel, there will be considerable increase in overall execution speed. In actual designs, algorithm will be divided in to parallel blocks and will be executed simultaneously. In a nutshell, the significant increase in processing speed is the major motivation behind hardware image processing for moving object tracking. If the processing time is less, the power consumption also will be reduced. Hence it can be concluded that hardware image processing for proposed systems give better performance in time critical applications. In the current scenario, most of the image processing algorithms are running in a sequential environment. Hence a research in FPGA based image processing algorithms has greater significance and scope in time critical applications.

Nowadays, Robots are used as mobile service robots in indoor environments such as houses, offices, hospitals. A number of sensors are used widely in order to detect obstacles in natural environments. However, most of these sensors are too expensive to apply for low-cost service robots like vacuum cleaning robots or guide robots. Hence, visual navigation takes much attention after web cameras were introduced a few years ago since its cost is attractive comparing with the previous sensors [2] (Nguyen Xuan Dao). Thus we can use Visual navigation using the cameras. Cameras are commonly used in the surveillance; these cameras can be used for determining the position of indoor objects.

1.2 Problem Formulation

With the advent of Micro-controllers, it is possible to design embedded image processing systems, which are portable, less power and time consuming. In micro-controller/DSP processors, the algorithm execution is sequential in nature. The speed of execution is greatly increased in advanced processors, which makes use of pipelined and super-scalar architectures. The advanced processors in corporates parallelism at instruction level, but the overall execution of the algorithm will be sequential in nature. Thus a micro controller based system cannot effectively utilize the inherent parallelism involved in most of the image processing algorithms. This imposes a limit on maximum processing rate. Thus such devices are not a suitable candidate for time critical applications.

Field Programmable Gate Arrays (FPGA) on the other hand gives a platform for parallel execution. In an FPGA based design, a different hardware block executes the sequences of an algorithm in parallel, and thus provides quick response and high frame rate. Since the overall operations are performed in less number of clock cycles, the power consumption will be reduced considerably, compared to micro-controller/DSP-processor based designs.

Generally GPS system is used for both indoor and outdoor navigation. But GPS system has some limitations, first is it requires costly infrastructure like navigation satellite and GPS module. Here navigation satellite is used as position monitoring equipment. We can use indoor surveillance camera as position monitoring equipment for indoor visual navigation application.

In indoor environments such as houses, offices, hospitals, mobile robots have to be equipped with a capability to navigate in dynamic environments to execute a given task while avoiding obstacles. A number of sensors such as laser finders, ultrasonic sensors, stereo-camera-based range sensors, and etc. are used widely in order to detect obstacles in natural environments. However, most of these sensors are too expensive to apply for low-cost service robots. Thus we can use low cost camera for indoor robot navigation using the visual navigation.

II. Methodology

This Paper proposes the novel indoor navigation system for indoor service robots. In this project mobile service robot considered as any general robot with less commutation or processing power and less number of sensors. Thus proposed navigation system can be used for any domestic simple robots.

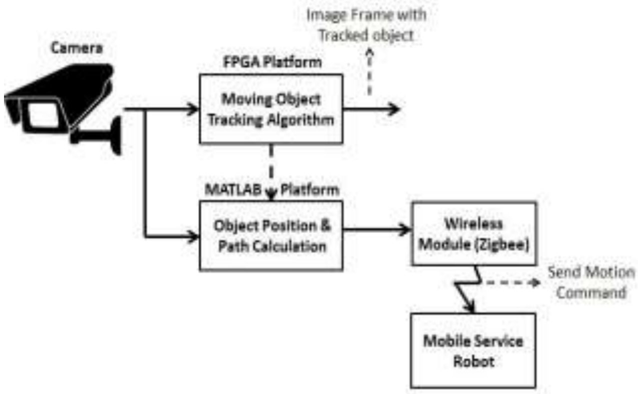


Fig. 1: System Block Diagram

As shown in fig.1. the camera is fixed at the some point in the indoor environment. This camera is used for both surveillance and the position monitor equipment for indoor navigation. Here we developed moving object tracking module in using the FPGA, the image frames captured by the camera are then given to the FPGA implementation for moving object tracking. The moving object algorithm is used for the extracting the image coordinates of the moving object which helps to calculating the center of the object. The out of FPGA platform is Image Frame with tracked object. At the MATLAB platform object position and Path calculation is done, also the motion command is also generated by the MATLAB algorithm. The motion command generated by MATLAB platform is send to the mobile robot through the wireless module. Motion commands used for moving the robot through the pre calculated path. MATLAB GUI is used for the representing the position of the moving object. Here wireless module is used for the communication and sending position coordinate as well as motion command to the mobile robot.

III. Intoduction To Basic Object Tracking

3.1 Basics Object Tracking

Object tracking is the process of locating a moving object in time using a camera. The algorithm analyses the video frames and outputs the location of moving targets within the video frame.

A few examples of established motion models are:

- To track objects in a plane, the motion model is a 2D transformation of an image of the object (the initial frame)
- When the target is a 3D object, the motion model defines its aspect depending on its 3D position and orientation
- The image of deformable objects can be covered with a boundary box, the motion of the object is defined by the position of the nodes of the bounding box.

The role of the tracking algorithm is to analyze the video frames in order to estimate the motion parameters. These parameters characterize the location of the target. They help identify several other factors such as average speed, number of direction changes, total time in motion and also information about the shape and size of the target.

3.*2 Methods of Implementation*

The two major components of a visual tracking system:

- Target Representation and Localization
- Filtering and Data Association

Target Representation and Localization is mostly a bottom-up process. Typically the computational complexity for these algorithms is low. The following are the common Target Representation and Localization algorithms:

Blob tracking: Segmentation of object interior (for example blob detection, block-based correlation).

- Mean-shift tracking: An iterative localization procedure based on the maximization of a similarity measure.
- Contour tracking: Detection of object boundary (e.g. Active Contours, Watershed Algorithm)
- Visual feature matching: Registration

Filtering and Data Association is mostly a top-down process, which involves incorporating prior information about the scene or object, dealing with object dynamics. The computational complexity for these algorithms is usually much higher. The following are some common Filtering and Data Association algorithms:

- Kalman filter: An optimal recursive Bayesian filter for linear functions and Gaussian noise.
- Particle filter: Useful for sampling the underlying state-space distribution of non-linear and non-Gaussian processes.

IV. Image Processing System For Moving Object Tracking

4.1 System Environment

Our aim is to work in an indoor environment. An indoor environment is one which has no artificial blue/green lights. As this environment requires the need to distinguish the objects of interest from any other objects that may be present within the frame. This limitation may be overcome by restricting the target objects to saturated and distinctive colors to enable them to be distinguished from the unstructured background. Augmenting the unstructured environment with 11 structured colors in this way is a compromise that enables a much simpler segmentation algorithm to be used. Another method to maintain the color distribution is to keep the background environment a constant. This way, only the target is in motion and the system is able to track its motion in a 2- D frame.

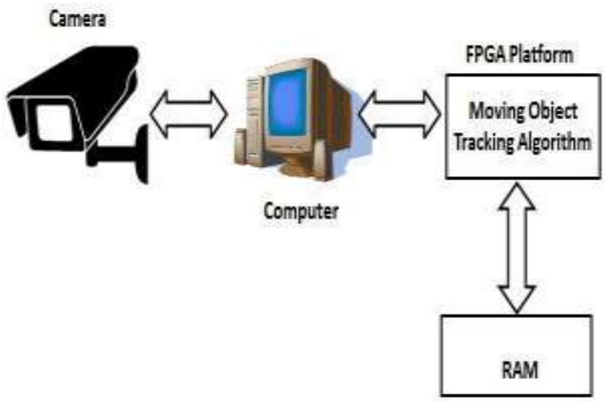


Fig.2: Layout of the Image Processing System

4.2 Image Acquisition

The image capture is performed using a color video camera which produces stream of RGB pixels. The circle camera is mounted at top of the room with a fixed Background that contains the object to be tracked. The temporal resolution requirements of the application have to be considered. We require a lower resolution as it will have a significantly higher acquisition rate for the observation of faster events. The size of the video frame is set to 460x819 pixels. The video obtained is read in the computer using MATLAB. The software processes the entire video and converts it into Image frames at the rate of 3 to 6 frames per second. Depending on the accuracy required and computational capability of the System, the frames can be interlaced [3][4][5] (Jung Uk cho; 2007, Rao Sandeep; 2012, Pandey Monoj; 2013).

4.3 Frame Generation

The MATLAB program reads video from the camera and converts it to frames. The frames are produced at the rate of 3 to 6 frames per second. These frames are then resized to 460 x 819 px size and stored in the variable used in MATLAB. Following MATLAB code shows Frame Generation.

4.4 Algorithm Design for Object Recognition

After the images are captured by the digital camera and stored in the memory unit, the location data (Center Coordinated of Object) are then extracted. Fig.3 shows a flowchart for the recovery of a location code from the captured images by using the image-processing algorithms developed [3][4][5] (Jung Uk cho; 2007, Rao Sandeep; 2012, Pandey Monoj; 2013). The Following MATLAB code shows object recognition algorithm.

4.5. Object Tracking

4.5.1 Optimal Frame Rate

The algorithm previously described helps identify the object and gives us information about its shape and size. Now in order to track it, we must select the frames acquired from the video. The frames are then fed into the MATLAB program at the rate of 3 frames per second. This is under the impression that the rate that we choose contains the complete motion of the object. The optimal frame rate considered is 1 frame per second. Further complexity is reduced if we alter the input frame rate to 3 frames per second.

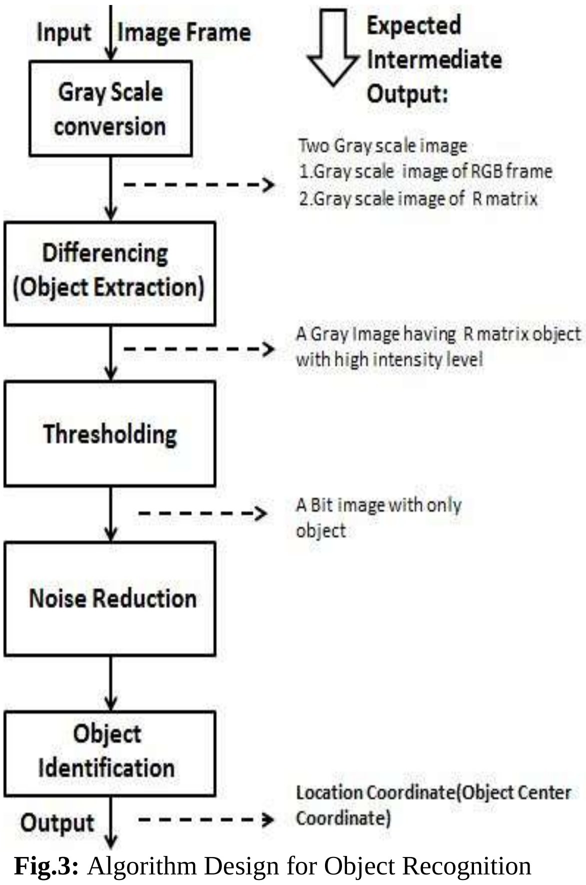
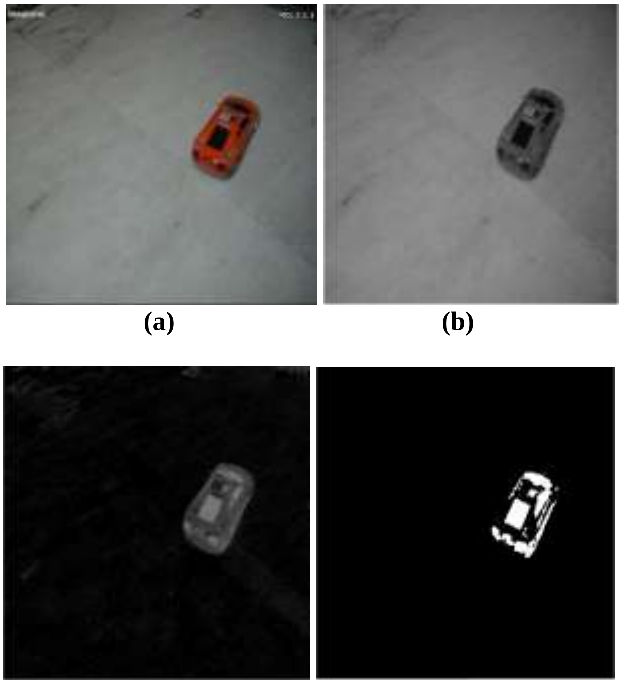


Fig.3: Algorithm Design for Object Recognition

4.5.2 Determining the Objects Position

In most applications, the center of gravity is used for tracking the target as it is a geometric property of any object. The center of gravity is the average location of the weight of an object. It can be used to describe the motion of the object through space in terms of the translation of the point of the object from one place to another.

Another method for determining the object position is the calculating the center of the object that is centroid of the object. The following MATLAB code gives Centroid of the detected object.



(a) **(b)**

(c) **(d)**

Fig.4: MATLAB simulation for moving object tracking (a) Orignal image (b) Gray image (c) after differencing (d) Thresholding.

V. FPGA For Object Tracking Algorithm

5.1 The Advantage of Using FPGAs

Image processing is difficult to achieve on a serial processor. This is due to the large data set required to represent the image and the complex operations that need to be performed on the image. Consider video rates of 25 frames per second, a single operation performed on every pixel of a 768 by 576 color image (Standard PAL frame) equates to 33 million operations per second. Many image processing applications require that several operations be performed on each pixel in the image resulting in an even large number of operations per second. Thus the perfect alternative is to make use of an FPGA [1][3]-[5] (Pujari Shashank; 2008, Jung Uk cho; 2007, Rao Sandeep; 2012, Pandey Monoj; 2013, M.Sreejith; 2014, Sofia Nayak; 2015). Continual growth in the size and functionality of FPGAs over recent years has resulted in an increasing interest in their use for image processing applications. The main advantage of using FPGAs for the implementation of image processing applications is because their structure is able to exploit spatial and temporal parallelism. FPGA implementations have the potential to be parallel using a mixture of these two forms. For example, the FPGA could be configured to partition the image and distribute the resulting sections to multiple pipelines all of which could process data concurrently. Such parallelization is subject to the processing mode and hardware constraints of the system.

5.2Hardware Constraints

There are three modes of processing: stream, offline and hybrid processing. In stream processing, data is received from the input device in a raster nature at video rates. Memory bandwidth constraints dictate that as much processing as possible can be performed as the data arrives. In offline processing there is no timing constraint. This allows random access to memory containing the image data. The speed of execution in most cases is limited by the memory access speed. The hybrid case is a mixture of stream and offline processing. In this case, the timing constraint is relaxed so the image is captured at a slower rate. While the image is streamed into a frame buffer it can be processed to extract the region of interest. This region can be processed by an offline stage which would allow random access to the regions elements [6] (M.Sreejith; 2014).

5.3Timing Constraints

If there is no requirement on processing time then the constraint on timing is relaxed and the system can revert to offline processing. This is often the result of a direct mapping from a software algorithm. The constraint on bandwidth is also eliminated because random access to memory is possible and desired values in memory can be obtained over a number of clock cycles with buffering between cycles. Offline processing in hardware therefore closely resembles the software programming paradigm; the designer need not worry about constraints to any great extent. This is the approach taken by languages that map software algorithms to hardware. The goal is to produce hardware that processes the input data as fast as possible given various automatic and manual optimization techniques [6] (M.Sreejith; 2014).

5.4Bandwidth constraints

Frame buffering requires large amounts of memory. The size of the frame buffer depends on the transform itself. In the worst case (rotation by 90º, for example) the whole image must be buffered. A single 24-bit (8-bits per color channel) color image with 768 by 576 pixels requires 1.2 MB of memory. FPGAs have very limited amounts of on-chip RAM. The logic blocks themselves can be configured to act like RAM, but this is usually an inefficient use of the logic blocks. Typically some sort of off-chip memory is used but this only allows a single access to the frame buffer per clock cycle, which can be a problem for the many operations that require simultaneous access to more than one pixel from the input image. For example, bilinear interpolation requires simultaneous access to four pixels from the input image. This will be on a per clock cycle basis if real-time processing constraints are imposed [6] (M.Sreejith; 2014).

5.5 Implementation of object tracking Algorithm

Xilinx ISE Design Suite-14 software platforms are used for
design object tracking algorithm; VHDL language used for writing algorithm. Image of size m x n is input to system; object detected image of same size obtained at output. Fig.3 illustrates object detection algorithm 24 bit RGB pixel value of input image is given as input to module. This module contains object recognition and object position calculation block. Modules are synchronies with input clock to system.

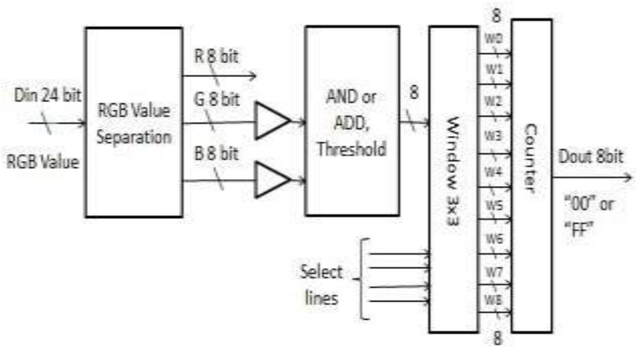


Fig. 5. Architecture of proposed FPGA module

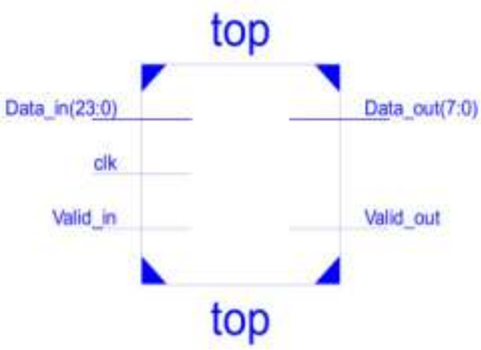


Fig. 6. RTL Schematics of moving object tracking module

5.6 Test bench

To test correctness of VHDL code a special type of code called test bench code is used. Xilinx provide feature called simulation which takes valid input and shows how it would work in hardware. To simulate design, both the design under test (DUT) or unit under test (UUT) and the stimulus provided by the test bench are required. The test bench is usually written in the same language (VHDL or Veri log) than entity under test. Test bench has three main purposes to generate stimulus for simulation, to apply this stimulus to the entity under test and to collect output responses. In VHDL test bench it is possible to read data from text file and write data in to text file. To obtained raw image data for processing, MATLAB software is used. Program written in MATLAB writes RGB pixel values of image into text file. Input text file contains data matrix of size m x n obtained from m x n size gray scale image using MA TLAB software. Data written in text file is of type integer. Program written in test bench read pixel data from input text file and convert integer values to bit vector of 24 bit length which given as input to unit under test for object detection process. Output of unit under test i.e. detection process is bit vector of length 8 bits, which again converted in to integer type. Output of simulation is collected in output text file in form of edge detected data matrix which is of type integer. For converting data matrix to gray scale image, content from output text file is loaded to MA TLAB software to obtained image from edge detected data matrix. Fig. 7 shows a test design flow to check correctness of written VHDL code for edge detection at software platform.

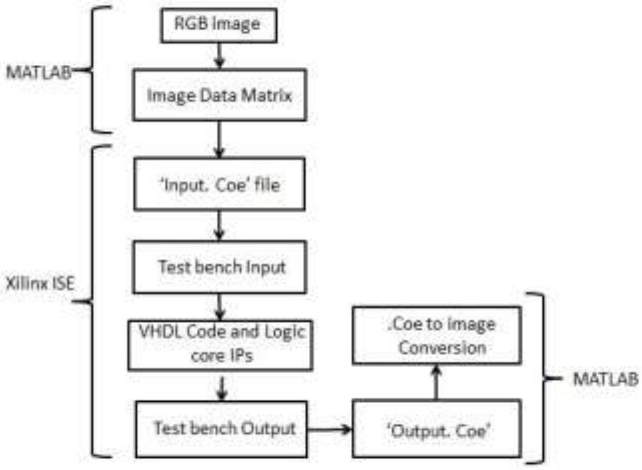


Fig. 7. Test Desgin Flow

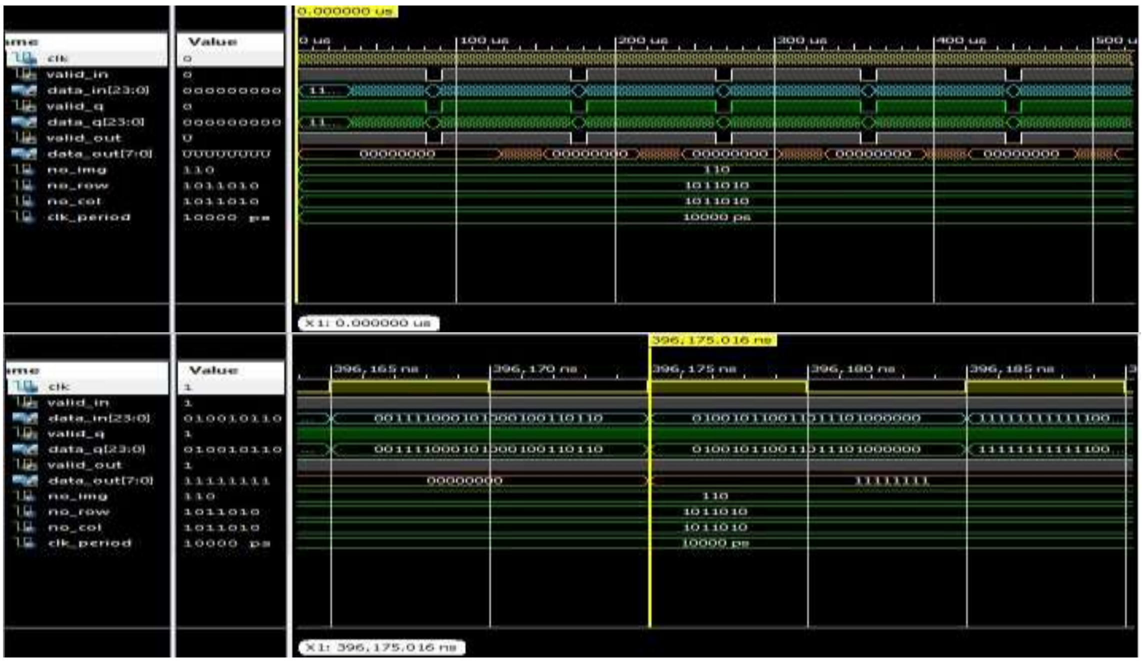


Fig. 9. ISim Simulation

5.7 Experimental result

Image of size 128x128 is used for experiment where number of rows are 128 and number of columns are 128. Xilinx FPGA device XC6VLX240T of family Virtex-6, is used as target device. To meet a real-time requirements; processing on single frame should be completed in desire time. For typical CMOS digital sensor camera maximum frame rate is 30 frame per second and maximum pixel clock frequency for which pixel data outs are valid is 66 MHz.

(a) (b)

Fig. 8 Experiment result (a) Orignal image (b) object detected/ tracked image

Device Utilization Summary (estimated values)			
Logic Utilization	Used	Available	Utilization
Number of Slice Registers	31	301440	0%
Number of Slice LUTs	34	150720	0%
Number of fully used LUT-FF pairs	31	34	91%
Number of bonded IOBs	19	600	3%
Number of BUFG/BUFGCTRL/BUFHCEs	1	176	0%

Table 1. Device Utilization Summary

VI. Path Calculation Algorithm And Motion Command

Path calculation algorithm at MATLAB platform is used for the calculating different paths in indoor environment from which robot can move in indoor environment. Path calculation algorithm also used for calculate the shortest path from with robot can navigate.

6.1 Path Calculation Algorithm

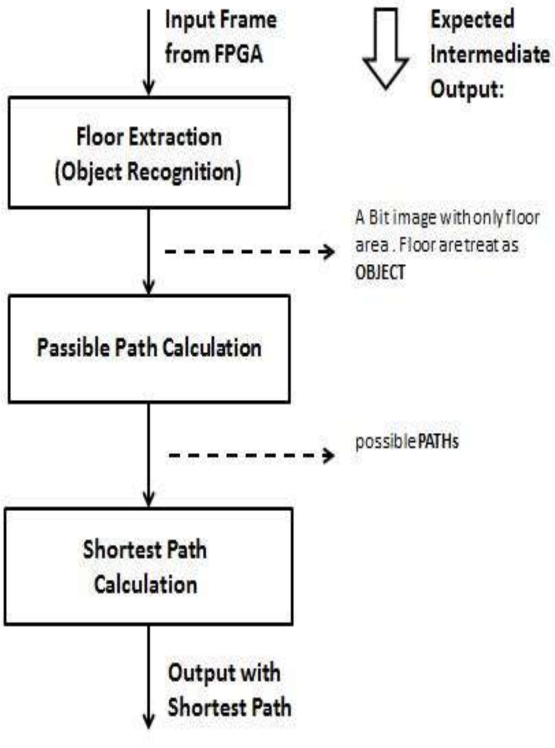


Fig.9. Flowchart of Path calculation algorithm

6.2 Floor Extraction (Object Recognition)

In indoor environment floor color properties are different from the other object. We can use theses floor properties for separating the floor area from the other object.

Here, Object Recognition algorithm is used for the floor extraction. This algorithm is explained priviously Instead of moving object as target object here floor area is used as target object. In the MATLAB simulation test image used as input image which is top view of the room taken camera mounted at center top of the room.

6.3 chessboard pattern method

In this method 10 x 10 check pattern is drawn over the image. If one of the obstacles is intersecting the checks lines then that check region is removed and remaining checks lines are treaded as virtual movable paths or possible paths

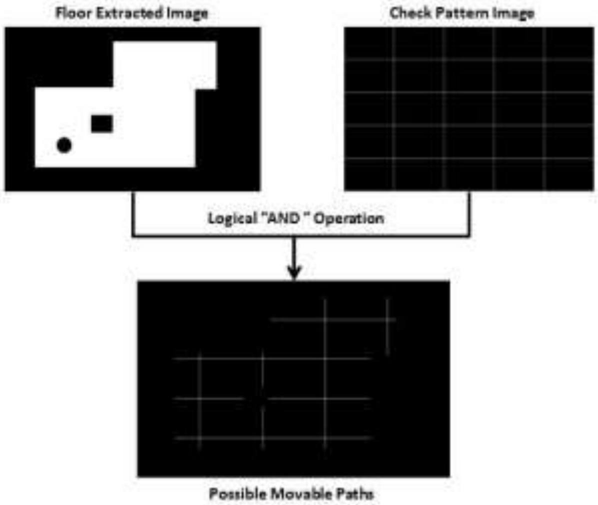


Fig.10. Chessboard pattern method; white area in first image floor area.

6.4 Shortest Path Calculation

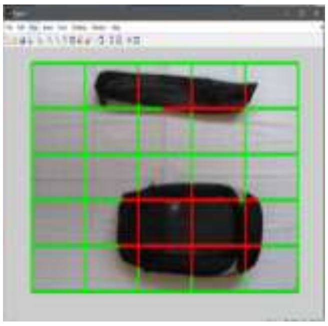
Here the shortest path is calculated between the starting coordinate and destination coordinates, where starting coordinate is nothing but position coordinate of moving object that is calculated during object tracking. Destination point is selected by the user. In shortest path calculation we used Geodesic Distance Transform for finding the minimum distance path [7] (Gonzalez R.C; 2007). In Geodesic Distance Transform, regions where skeleton image is true represent valid regions that can traverse in the computation of the distance transform. Then the valid distance region is dilated to highlight the shortest path.

6.5 Motion Command Generation

Motion commands are used for moving robot to destination point, also it used for make robot to follow predefined path. Motion command generation depends on the shortest path calculation that is path coordinates and robot current position coordinate. Comparing the path coordinate and robot motion coordinate motion commands are generated. These motion commands are sending to the robot through the wireless media using the zigbee module serially.

6.5 Experimental result

(a) (b)



(c)

(d) (e)

Fig.11. Path finding results (a) Original image (b) Floor extracted image (c) Possible movable paths shown in green (d) (e) shortest path output with two different destination coordinate

VII. Conlusion

Reconfigurable device such as FPGA is very efficient for image processing. In the current scenario, most of the image processing algorithms are running in a sequential environment, but FPGA offers highly flexible, parallel processing. Hence a research in FPGA based image processing algorithms has greater significance and scope in time critical applications.

Moving object tracking is one of the fundamental components of computer vision; it can be very beneficial in applications such as unmanned aerial vehicle, surveillance, automated traffic control, biomedical image analysis, intelligent robots etc. Object recognition and Localization can be used for moving object tracking using the objects visual properties. Here moving object tracking efficient

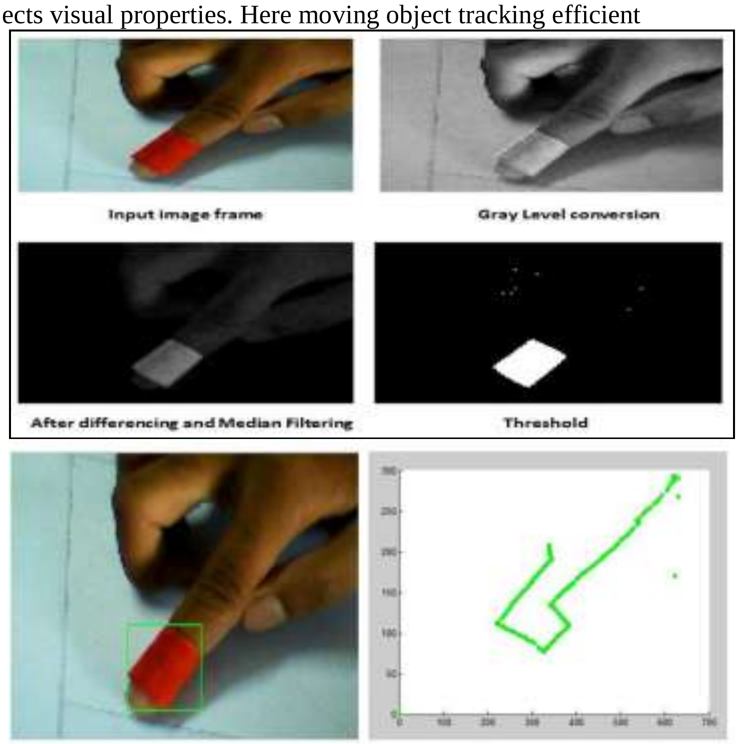


Fig.12. MATLAB simulation for moving object tracking

when we use the high frame rate cameras for image acquisition. Continuous Plot of path traced by the moving object is depends upon the frame rate of the camera.

Finally we can conclude that FPGA is ideal choice for implementation of moving object tracking algorithm based visual navigation.

Acknowledgement

The authors wish to thank Department of Technology, Shivaji University, Kolhapur and Mtech Coordinator Prof, P. C. Bhaskar for the support to providing developing tools, laboratory infrastructure in cayying out this work.

References

[1]. Pujari Shashank, Bhandari Sheetal, Chandak Sudarsan (2008), "FPGA Controlled Vision System for Survillance Robot (UAV)", CSI Communication, Robotics, Nov. 2008, Vol32

[2]. Nguyen Xuan Dao, Bum-Jae You, Sang-Rok Oh, "Visual navigation for indoor mobile robots using a single camera".

[3]. Jung Uk cho, Seung hun jin, Xuan Dai Pham, Dong kyun Kim, and jae wook Jeon (2007), "FPGA Based Real Time Visual Tracking System using Adaptive color Histograms", IEEE, 2007.

[4]. Rao Sandeep, Natarajan Aranind, Moorthi S.and Selvan M. P.(2012), "Real Time Object Tracking in a video Stream using Field Programmable Gate Array", IEEE, 2012.

[5]. Pandey Monoj , Bprgohain Dorothi, Baruah Gargi (2013), "Real Time Object Tracking : Simulation and Implementation on FPGA Based Soft Processor", ICSSITE, 2013.

[6]. M. Sreejith (2014), "Development of an FPGA based Image processing intellectual Property core" May 2014.

[7]. Gonzalez R. C., Woods R. E.(2007), Digital Image Processing. 3rd ed. Prentice Hall, 2007, pp. 187-190.

[8]. Baumela L. and Maravall D. (1995) "Real-time target tracking," Aerospace and Electronic Systems Magazine, IEEE, vol. 10, no. 7, pp. 4-7, 1995.

[9]. Chang Chi-Jeng, Hsiao Pei-Yung, Huang Zen-Yi (2006). "Integrated Operation of Image Capturing and Processing in FPGA", IJCSNS International Journal of Computer Science and Network Security, VOL.6 No.1A, pp 173-179.

[10]. Chaple Girish , Daruwala R. D.(2014), "Design of Sobel Operator based Image EdgeDetection Algorithm on FPGA" International Conference on Communication and Signal Processing, April 3-5, 2014, India..

[11]. Chunzhong Chen, Baofeng Guo, Xiangwei Wu (2014) "An Edge Detection Method for Hyperspectral Image Classification Based on Mean Shift", The 2014 7th International Congress on Image and Signal Processing

[12]. Crookes D., Benkrid K., Bouridane A., Alotaibi K., and Benkrid A.(2000), "Design and implementation of a high level programming environment for FPGA-based image processing", Vision, Image and Signal Processing, IEE Proceedings, vol. 147, Issue: 4 , Aug, 2000, pp. 377 384.

[13]. Fire Bird V research platform hardware and software manual.

[14]. Hayato Hagiwara, Kenichi Asami, and Mochimitsu Komori (2012), "Real-time Image Processing System by Using FPGA for Service Robots", IEEE Global Conference on Consumer Electronic 2012.

[15]. Hong C.S,. Chung S.M, Lee J.S. and Hong K.S. (1997), "A Vision- Guided Object Tracking and Prediction Algorithm for Soccer Robots", IEEE Robotics and Automation Society, Vol. 1 pp: 346-351.

[16]. Jadhav Suhas, Narvekar Rohit, Mandawale Ajay, Elgandelwar Sachin (2015), "FPGA Based Object Tracking System", Fifth International Conference on Communication Systems and Network Technologies 2015.

[17]. Johnston Christopher T., Gribbon Kim T, Bailey Donald G. (2005) "FPGA based Remote Object Tracking for Real-time Control", 1st International Conference on Sensing Technology November 21-23, 2005 Palmerston North, New Zealand

[18]. Kilmartin L., O Conghaile M. (1999), "Real Time Image Processing Object Detection and Tracking Algorithms", Proceedings of the Irish Signals and Systems Conference, NUI, Galway, June 1999, pp. 207- 214.

[19]. Nayak Sofia, Pujari Shashank (2015), "Moving Object Tracking Application: FPGA And Model Based Implementation Using Image Processing Algorithms" International Conference on Computing Communication Control and Automation 2015.

[20]. Rodrigues Ingrid Nunes, Charles Luiz Silva de Melo, Vanessa Mello da Frota Botinelly and Jozias Parente de Oliveira, "FPGA hardware architecture with parallel data processing to detect moving objects using the background image subtraction technique".

[21]. BOOK REFERECES

[22]. Digital Image Processing 3rd ed. - R. Gonzalez, R. Woods

IOSR Journal of VLSI and Signal Processing (IOSR-JVSP) is UGC approved Journal with Sl. No. 5081, Journal no. 49363.

T. D. Magdum . "FPGA Based Moving Object Tracking For Indoor Robot Navigation." IOSR Journal of VLSI and Signal Processing (IOSR-JVSP) , vol. 7, no. 5, 2017, pp. 12–22.