IOSR Journal of Electronics & Communication Engineering (IOSR-JECE)
ISSN(e) : 2278-1684 ISSN(p) : 2320-334X, PP 16-22
www.iosrjournals.org

A New Object Detection and Tracking Using FPGA

Ahinus H[1] , Gopalakrishnasamy M E[2]

[1] *Department of Electronics and Communication ,Maharaja Engineering College,Avinasi, Coimbatore*

[2] *Department of Electronics and Communication ,Maharaja Engineering College,Avinasi, Coimbatore*

ahinus@gmail.com.

Abstract

According to the result of moving object detection research on video sequences, this paper proposes a method to detect moving object based on background subtraction. Computational using local variable such as Lucas Kanade algorithm does not provide a good segmentation which indirectly affects the pattern of the optical flow obtained. Currently, both the market and the academic communities have required applications based on image and video processing with several real-time constraints. On the other hand, detection of moving objects is a very important task in mobile robotics and surveillance applications. In order to achieve an alternative design that allows for rapid development of real time motion detection systems, this paper proposes a hardware architecture based on the background subtraction algorithm, which is implemented on FPGAs (Field Programmable Gate Arrays).Here we will calculate total power consumed by our hardware .

Keywords*– noise removal, image segmentation,FPGA*

1. INTRODUCTION

In this project we propose to use Image Processing algorithms for the purpose of Object Recognition and Tracking and implement the same using an FPGA. In today’s world most sensing applications require some form of digital signal processing and these are implemented primarily on serial processors. While the required output is achievable, it can be beneficial to take advantage of the parallelism, low cost, and low power consumption offered by FPGAs (Spartan 3). The Field Programmable Gate Array (FPGA) contains logic components that can be programmed to perform complex mathematical functions making them highly suitable for the implementation of matrix algorithms.

Image edge detection is an important technique in the area of image processing with wide applications in Medicine, Remote sensing, military to mention a few. There are conventional as well as improvised edge detection algorithms depending on the application. The choice of the technique in most cases depends on the application and image in question rather than a generalized method.. The objective of this paper is to realize the edge detection algorithm on a FPGA (Field Programmable Gate Array). The segmentation processor also has the capabilities to perform other tasks so that it could mimic an image processor. FPGA implementation renders it more useful for real time applications.

Image segmentation is a technique and process which divide the image into different feature of region and extract out the interested target. To illustrate the level of the image segmentation in image processing, we have introduced "image engineering" concept ", it bring the involved theory, methods, algorithms, tools, equipment of image segmentation into an overall framework. With the improvement of computer processing capabilities and the increased application of color image, the image segmentation are more and more concerned. This article proposes a image segmentation method based on the traditional seed region growing algorithm. The individual frames acquired from the target video are fed into the FPGA. These are then subject to segmentation, thresholding and filtering stages. Following this the object is tracked by comparing the background frame and the processed updated frame containing the new location of the target. The results of the FPGA implementation in tracking a moving object were found to be positive and suitable for object tracking.

2. OBJECT TRACKING AND SYSTEM DESIGN

A. Basic 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. Our aim is to work in an unstructured environment. An unstructured environment is one which has no artificial blue/green screen. This provides greater system flexibility and portability but can make reliable segmentation more difficult. 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 structured color 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.The image capture is performed using a color video camera which produces a stream of RGB pixels. The software processes the entire video and converts it into Image frames at the rate of 10 frames per second. Depending on the accuracy required and computational capability of the System, the frames can be interlaced.

B Frame Generation

From the video frames are created. The frames are produced at the rate of 10 frames per second. Consider a 10 second video, a total of 100 frames will be produced in RGB format. These frames are then stored as individual bitmap files (total of 100 files). The bitmap files are arranged in the order of their occurrence in the video. The first frame is selected as the Base – Background Frame. The remaining bitmap files are used for the process of Object Recognition and Tracking.

C. Background and Object Identification

It is important that the object needs to be differentiated from the background. The color elements must be eliminated and the recognition is done in gray scale. With a still environment, the background frame is selected as the first frame. Considering the 10 second video, the 50th frame is randomly selected as the Object frame – these two frames form the basis for Object Recognition. It gives information about the shape and size of the object.

2.1 Algorithm Design For Object Recognition

The following modules make up the Object Recognition stage: Grayscale Conversion, Delta Frame Generation, Thresholding, Noise Filtering and Image Enhancement. Fig: 1 shows the Algorithm Flow.

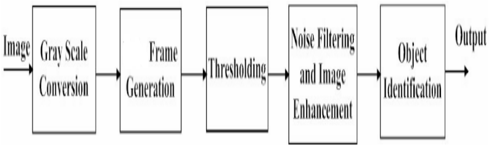


Fig 1: Object Recognition Algorithm Flow

2.1.1 Grayscale Conversion

The bitmap files have been generated and the Background and Object frame have been selected. These files are present in RGB format at a resolution of 640x480 pixels. These frames are then converted to grayscale within a range of 0-255. This reduces the coherent effect of the environment and allows us to easily separate the object from the background.

2.1.2 Delta Frame Generation

Once the Gray Scale Conversion has been completed, the respective frames are subtracted from one another. The resulting frame is called the Delta Frame. This method of image subtraction eliminates the background and brings the object into focus, giving us information about its shape and size. The Delta frame also reduces the number of pixels that the system will have to process.

2.1.3 Thresholding

In order to further enhance the resolution of the delta frame Gray Scale Thresholding is done. The individual pixels in the grayscale image are marked as object pixels if their value is greater than some threshold value (initially set as 80) and as background pixels otherwise.

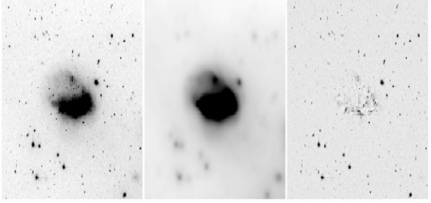


Fig 2: Gray Level Thresholding

In this case the object pixel is given a value of "1" while a background pixel is given a value of "0." The thresholding can also be made adaptive when a different threshold is used for different regions in the image. The initial threshold value is set by considering the mean or median value. The approach is justified if the object pixels are brighter than the background. Else an iterative method has been implemented to obtain the value.

The algorithm is as follows –

An initial random threshold (T) is chosen.

1. The image is segmented into object and background pixels using the above threshold. This creates two sets:

    1. $G1 = \{f(m,n):f(m,n)>T\}$ (object pixels)

2. $G2 = \{f(m,n):f(m,n) T\}$ (background pixels)

3. The average of each set is computed.

    1. $m1$ = average value of $G1$

    2. $m2$ = average value of $G2$

4. A new threshold is created that is the average of $m1$ and $m2$

    1. $T' = (m1 + m2)/2$

2.1.4 Noise Filtering

The median filter is normally used to reduce noise in an image. The median filter is considered to do a better job than the mean filter of preserving useful detail in the image. The filter considers each pixel in the image in turn and looks at its nearby neighbors to decide whether or not it is representative of its surroundings. It then replaces the pixel value with the median of the neighboring pixel values. The median is calculated by first sorting all the pixel values from the surrounding neighborhood into numerical ascending order and then replacing the pixel being considered with the middle pixel value. (If the neighborhood under consideration contains an even number of pixels, the average of the two middle pixel values is used.)

3. FPGA IMPLEMENTATION OF OBJECT TRACKING ALGORITHM

3.1 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. 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.

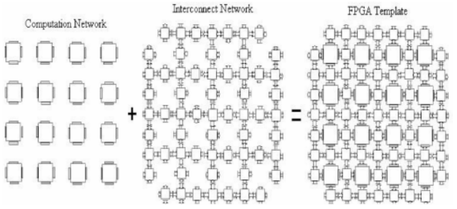


Fig 3: Programmable Logic Blocks of an FPGA

In Fig 3, an FPGA consists of a matrix of logic blocks that are connected by an interconnect network. Both the logic blocks and the interconnect network are reprogrammable allowing application specific hardware to be constructed, while at the same time maintaining the ability to change the functionality of the system with ease. As such, an FPGA offers a compromise between the flexibility of general purpose processors and the hardware-based speed of ASICs.

4. RESULT

The goal of image segmentation is to cluster pixels into salient image regions, i.e.,regions corresponding to individual surfaces, objects, or natural parts of objects. The hardware was realized on a Spartan-3 EDK Development Board. The processor was coded using C language and simulated using XILINX XPS 10.1.XILINX XPS cannot handle the standard image formats so the images were converted to ASCII text files using MATLAB. The ASCII text file was applied as vector to the hardware interface. The output files were similarly converted and viewed in MATLAB. The execution time for the entire program of edge detection for an image of size 128×128 is for few seconds. The region growing algorithm involves the technique called background subtraction. Background subtraction is subtracting the images from the background by using pixel values. Here we have system hardware and software, we can estimate its power consumption. In embedded systems (and not only) power is very important since they determine the cooling, power supply and battery lifetime. To get a rough estimate of our system's power, here used Xpower Analyzer tool that comes with Xilinx ISE, fig show total quiescent power consumed by our Spartan 3 FPGA.

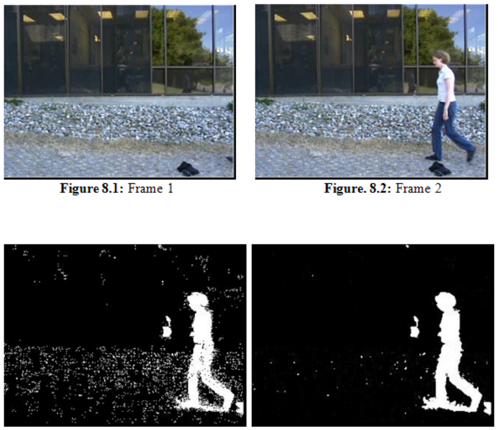
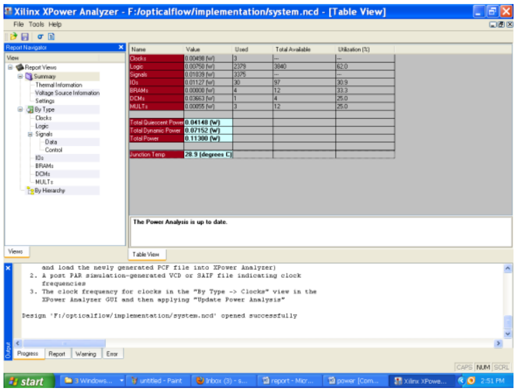


Figure 8.1: Frame 1 **Figure. 8.2:** Frame 2

Figure. 8.3:Object Tracked **Figure. 8.4**:Object Tracked(noise removed)

Fig:4 Object Tracking



Name	Value	Used	Total Available	Utilization (%)
Clocks	0.00498 (W)	3	---	---
Logic	0.00750 (W)	2379	3840	62.0
Signals	0.01039 (W)	3375	---	---
IOs	0.01127 (W)	30	97	30.9
BRAMs	0.00000 (W)	4	12	33.3
DCMs	0.03663 (W)	1	4	25.0
MULTs	0.00055 (W)	3	12	25.0
Total Quiescent Power	0.04148 (W)			
Total Dynamic Power	0.07152 (W)			
Total Power	0.11300 (W)			
Junction Temp	28.9 (degrees C)			

The Power Analysis is up to date.

```
      and load the newly generated PCF file into XPower Analyzer)
   2. A post PAR simulation-generated VCD or SAIF file indicating clock
      frequencies
   3. The clock frequency for clocks in the "By Type -> Clocks" view in the
      XPower Analyzer GUI and then applying "Update Power Analysis"

Design 'F:/opticalflow/implementation/system.ncd' opened successfully
```

Fig:5 Power Consumed by Spartan 3 FPGA

5. CONCLUSION:

In this paper, a real-time and accurate method for detecting moving body is proposed, based on background subtraction. In cognizance of the shortcomings and deficiencies in the traditional method of object detection, we establish reliable background model, use dynamic threshold method to detect moving object and update the background in real time. At last, we combine contour projection analysis with shape analysis to remove the shadow effect. Experiments show that the algorithm is fast and simple, able to detect moving body better and it has a broad applicability.

REFERENCES

[1] Y. Matsushita, E. Ofek, W. Ge, X. Tang, and H.-Y. Shum, “Full-frame video stabilization with motion inpainting,” *IEEE Trans. Patt. Anal. Mach. Intell., vol. 28, no. 7, pp.* 1150–1163, Jul. 2006.

[2] F. Barranco, J. Diaz, E. Ros, and B. del Pino, “Visual system based on artificial retina for motion detection,” *IEEE Trans. Syst., Man, Cybern., vol. 39, no. 3, pp.* 752–762, Jun. 2009.

[3] M. Anguita, J. Diaz, E. Ros, and F. J. Fernandez-Baldomero, “Optimization strategies for high-performance computing of optical-flow in general-purpose processors,” *IEEE Trans. Circuits Syst. for Video Technol., vol. 19, no. 10, pp.* 1475–1488, Oct. 2009.

[4] J. Diaz, E. Ros, F. Pelayo, E. M. Ortigosa, and S. Mota, “FPGA-based real-time optical-flow system,” *IEEE Trans. Circuits Syst. for Video Technol., vol. 16, no. 2, pp.* 274–279, Feb. 2006.

[5] Y. Matsushita, E. Ofek, W. Ge, X. Tang, and H.-Y. Shum, “Full-frame video stabilization with motion inpainting,” *IEEE Trans. Patt. Anal. Mach. Intell., vol. 28, no. 7, pp.* 1150–1163, Jul. 2006.