



Feature Extraction of Edge Detected Images

Mayuri R. Wankhade¹, Narendra M. Wagdarikar²

¹Department of Electronics & Telecommunication, Pune University, India

²Department of Electronics & Telecommunication, Pune University, India

<mailto:mayuri.wankhade07@gmail.com>; ¹ narendradsp@rediffmail.com; ²

Abstract— Nowadays, Digital image processing is an ever expanding and dynamic area with applications reaching out into our everyday life such as in medicine, space exploration, surveillance, authentication, automated industry inspection and in many more areas. For building such high speed systems, Edge Detection and Feature-Extraction, as an important process in motion recognition is required to be processed. Edge is a basic feature of an image. Edges define the boundaries between regions in an image by locating sharp discontinuities in pixel values, which helps with segmentation and object recognition. Image Edge detection significantly reduces the amount of data and filters out useless information, while preserving the important structural properties in an image. Since edge detection is in the forefront of image processing for object detection. This project describes Sobel Edge Detection algorithm and labelling algorithm for extracting features of images.

Keywords— Sobel Edge Detection, Two Pass connected component labelling, MATLAB/ Simulink, Virtex-5 (xc5vsx50t-1ff1136) FPGA

I. Introduction

Nowadays, in various applications such as video surveillance, automotive vehicle control, efficient human-computer interaction, remote gesture control and so on motion recognition is becoming increasingly important. For this applications high speed image processing can significantly provide much details of the motion, which is very useful for improving the speed and precision of the recognition. For the applications which are useful for our everyday life such as in medicine, space exploration, surveillance, authentication, automated industry inspection and in many more areas digital image processing is an ever expanding and dynamic area. For building such high speed systems, Motion Feature-Extraction is an important and computationally heavy process in motion recognition. Firstly edge detection is required to be done before the process of feature extraction of images. Application specific hardware offers much greater speed than a software implementation. There are two types of technologies available for hardware design. Full custom hardware design called as Application Specific Integrated Circuits (ASIC) and semi custom hardware device, which are programmable devices like Digital Signal Processors (DSP) and Field Programmable Gate Arrays (FPGA). ASIC design offers highest performance, but the complexity and the cost is very high and time taken to design the hardware

is also very high. DSP's are well suited to extremely complex math intensive tasks such as image processing but hardware design knowledge is still required and the learning curve is much lower than some other design choices. Field Programmable Gate Arrays are programmable devices. They are also called reconfigurable devices. Hardware design techniques such as parallelism and pipelining can be developed on a FPGA because of which speed of operation increases, so it will be beneficial in reduction of execution time. So FPGAs are ideal choice for implementation of image edge detection. Software implementation of edge detection and feature extraction carried out in MATLAB. Edge detection can also be implemented on Spartan 3, Spartan 6, Virtex5 FPGA boards. Edge detection is the process of classifying and locating discontinuities in digital image intensity. In many image processing algorithms edge detection is always first step as it reduces the amount of data and filters out useless information. In period of last three decades much literature is published on edge detection in which different edge detection algorithms such as sobel, Roberts, prewitt, laplacian, LOG, canny, Beamlet transform techniques and many more are present.

II. System Architecture

For feature extraction of images firstly edge detection of images and after that labeling algorithm is used. A system block diagram explains basic idea about implementation of the system.

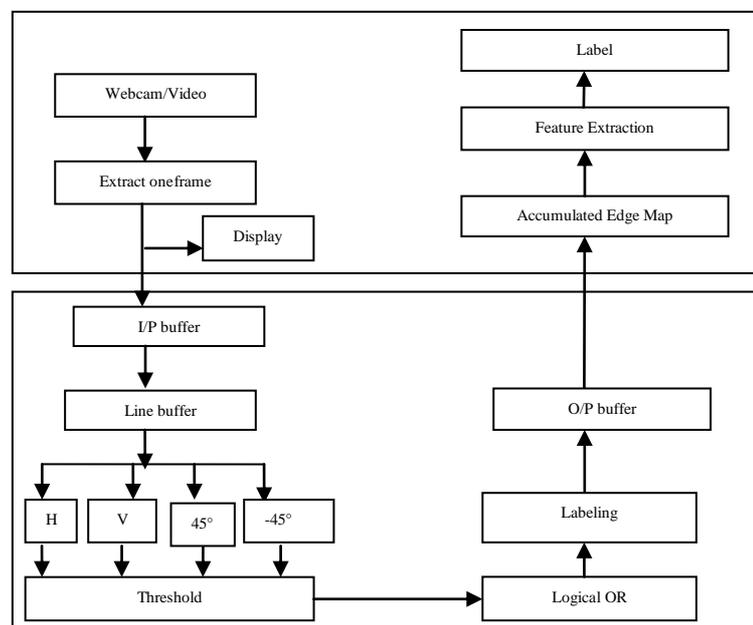


Fig.1 System Block Diagram

A. Edge Detection

Basic idea about hardware implementation of sobel edge detection algorithm on FPGA board is shown in Fig.2 The overall system consists of following blocks input image & image pre-processing, image edge detection process, Merged Significant Edge Map generation, Virtex5 FPGA board.

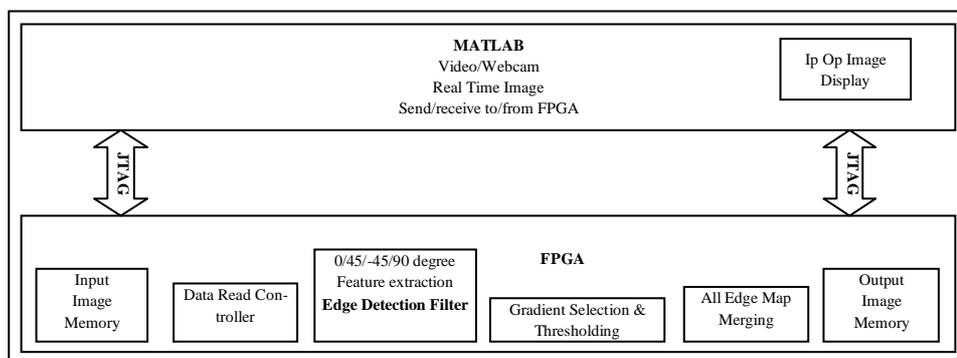


Fig.2 Block Diagram of Edge Detection

The block diagram shown above explains edge mapping and edge detection by hardware co simulation. By using FPGA and MATLAB edge detection and mapping take place. In this process firstly video frames are captured through the MATLAB and this video frames are captured via stored video or webcam. And after that this video frames are passed to share memory. During this, Image will be stored to FPGA memory and MATLAB Tool will manage automatically. After that input image data is read by using data read controller block. Then this image data will be passed to edge detection block. In edge detection filter block four different filters will work simultaneously. These four filters are the 0/45/-45/90 degree filters. Then output of these filters will be given to the next block that is Gradient Selection and Thresholding block. Therefore all the output of filters will be threshold. Now all edge map images will be combined as shown in above fig. by using edge map merging block and store in to memory. Here also Image will be read from FPGA memory and MATLAB Tool will manage automatically. Finally both input and output that is edge map frames will be displayed in real time in MATLAB.

B. Labeling algorithm

Labeling component is a commonly used technique for extracting high-order objects. Labeling Components are the basic ingredients of all high-order objects therefore a commonly used technique for extracting objects. Labels each component using a contour tracing technique. This method is based on the principle that a component is fully determined by its contours, just as a polygon is fully determined by its vertices. This method also provides a procedure for finding all component pixels. An image can be scan the same way as it would be encountered by a scanner, i.e., from top to bottom and from left to right per each line.

1) *Overview of Two Pass Connected Component Labeling Algorithm:* Algorithms discussed thus far are mostly suitable for all foregrounds or all background distinct pixels and their immediate neighbors. Here we are considering middle ground between the set of all foreground pixels and background pixels. This arises the general concept of connected components which is also called as objects in further discussion.

Two-pass algorithms: Many algorithms in this group operate in three distinct phases.

- (i) Scanning phase: In this phase, the image is scanned once to assign provisional labels to object pixels, and to record the equivalence information among provisional labels.
- (ii) Analysis phase: This phase analyzes the label equivalence information to determine the final labels.
- (iii) Labeling phase: This third phase assigns final labels to object pixels using a second pass through the image. The system flow diagram carries out the process of tracing and labeling of an object in given input image. This method uses multiple tracing. The flow chart given below explains the flow of connected component labeling algorithm. It gives the detail idea about working of algorithm.

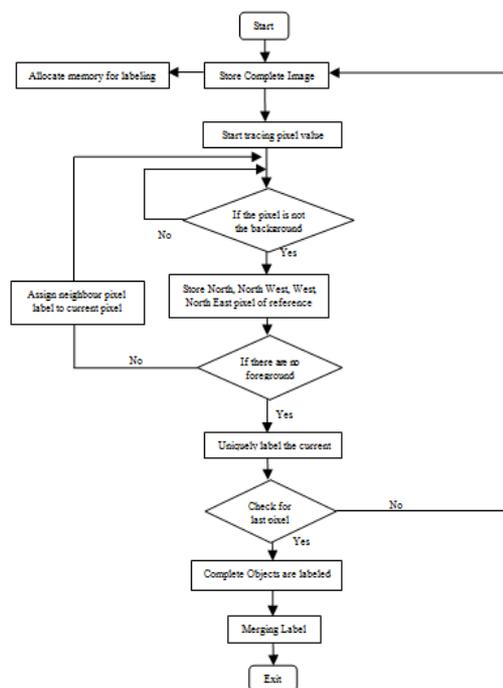


Fig. 3 Flow Chart of Labeling Algorithm

2) *Eight Connected Components Mask*: A graph, containing vertices and connecting edges, is constructed from relevant input data. The vertices contain information required by the comparison heuristic, while the edges indicate connected 'neighbour's. An algorithm traverses the graph, labeling the vertices based on the connectivity and relative values of their neighbours. 4connected or 8connected can be use for determining the connectivity as it is like medium, image graphs .Center Pixel at coordinates (x,y) has four neighbor pixels or they can be diagonal neighbors as shown in figure below.

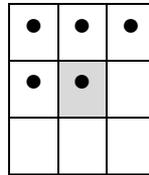
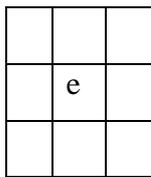


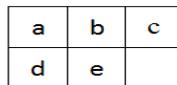
Fig. 4 Eight connectivity

This set of neighbors is denoted by $N_8(b)$. The four neighbors (x+1, y+1), (x,y+1), (x-1,y-1),and (x-1,y) are denoted by $N_D(c)$.If there exists above connected graph between the foreground pixels then they are said to be 8 connected. A connected component was defined in terms of path and path is decided by type of adjacency used.

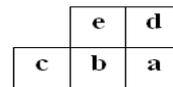
3) *Background*: For labeling connected components there are number of different approaches. The simplest approach repeatedly scans the image to determine appropriate labels until no further changes can be made to the assigned labels. A label assigned to an object pixel before the final assignment is called a provisional label. For a 2D image, a forward scan assigns labels to pixels from left to right and top to bottom. A backward scan assigns labels to pixels from right to left and bottom to top.



(a) 8-connected Neighbourhood mask



(b) Forward scan



(c) Backward scan

Fig. 5 The masks and the neighbourhood of pixel e. Notice that all the pixels in the forward and backward scan masks are in the neighbourhood of pixel b

Each time a pixel is scanned and to determine an appropriate label for the current pixel it's neighbours in the scan mask are examined. In the illustration shown in Fig.5 , the current pixel being examined is marked as e and the four neighbours in the scan masks are designated as a, b, c and d. The current pixel receives a new provisional label if there is no object pixel in the scan mask. On the other hand, if there are object pixels in the scan mask, the provisional labels of the neighbours are considered equivalent, a representative label is selected to represent all equivalent labels, and the current object pixel is assigned this representative label. A common strategy for selecting a representative is to use the smallest label.

An example of two pass algorithm is shown below:

1. The array shown below assign different pixel values to elements in the image. The 0 and 1 values written on the center of the elements in the following graph.

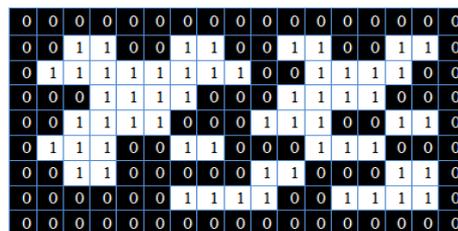


Fig. 6 Graph with different binary values for foreground and background elements.

2. After first pass labels are generated from "1, 2,...,7" values according to the flow diagram describe above. A total of 7 labels are generated in accordance with the conditions highlighted above. The graph generated after first pass is shown below.

3. After this, during second pass merging of labels is carried out. In this process the smallest value of label is assigns to the connected region and hence two distinct labels are generated as shown below.

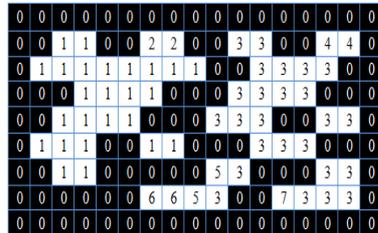


Fig 7 Labels generated after first pass

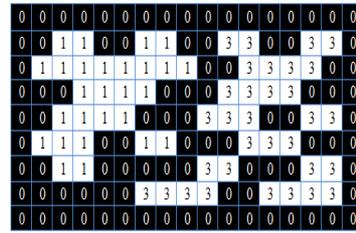


Fig 8 Labels generated after second pass

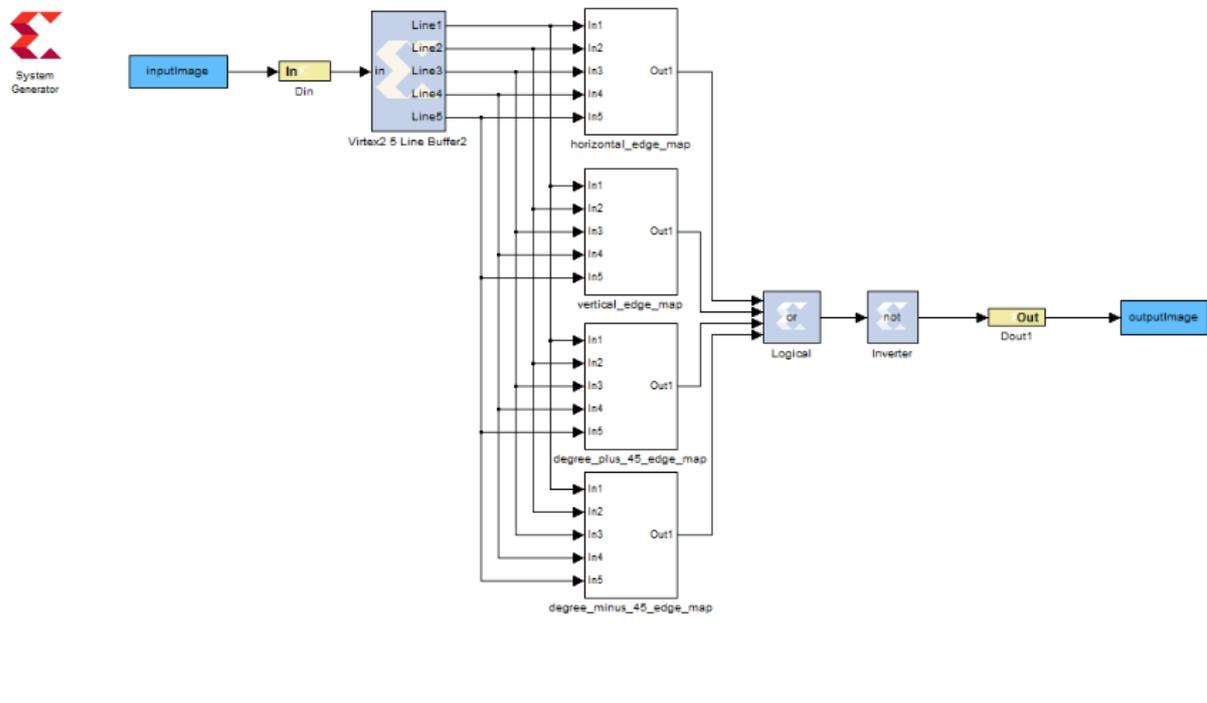
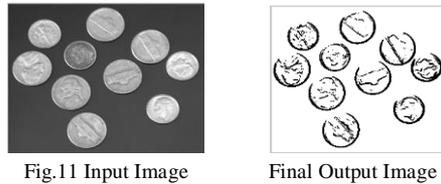
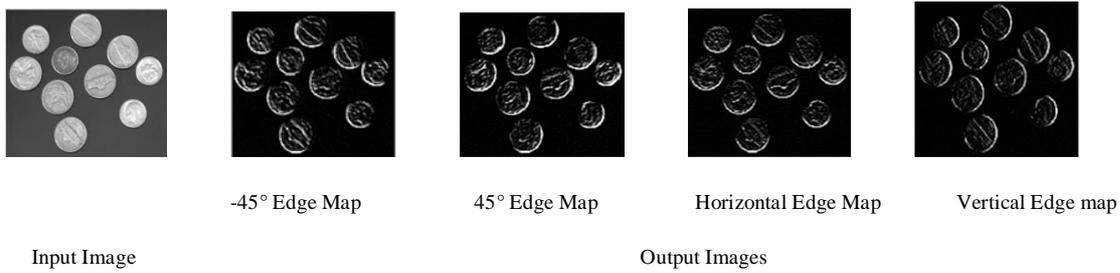


Fig.10 System generator model of Edge Detection algorithm



B) FPGA Results for Edge Detection

1) For Inbuilt Input Image from MATLAB:

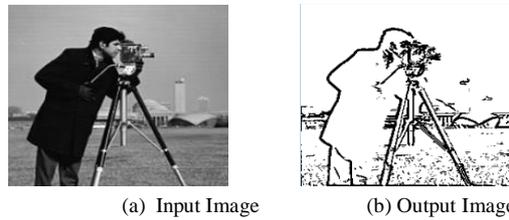


Fig. 11 FPGA Result for Inbuilt MATLAB Image of Cameraman

2) For Inbuilt Video:

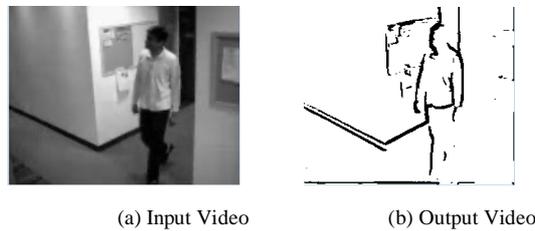


Fig.12 FPGA Result for Inbuilt MATLAB Video

3) For Real Time Input via Webcam:

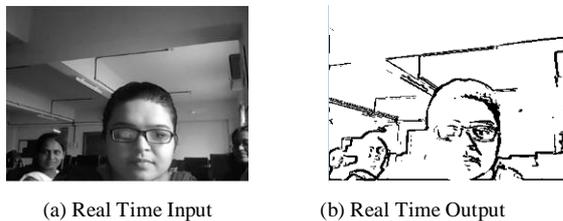


Fig.13 FPGA Result for Real Time Input via Webcam

C) Modeling of FPGA Implementation using Two Pass Connected Component Labeling Algorithm

By double-clicking on the System Generator icon in Matlab, the System Generator Library Browser opens. The Video and Image Processing toolbox from the library is used to design this model. From this toolbox, access the different elements used to design this model. The various blocks, elements and the sub-systems included in this model are described in detail below. Fig.14 shows the System generator model design of Two Pass Labeling algorithm based on connected component labeling.

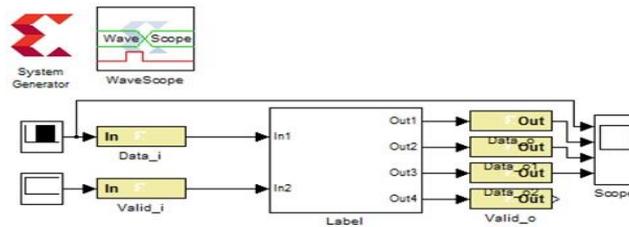


Fig. 14 System generator Model of Two Pass connected component Algorithm

1) *Assign Different labels with Different Colors:* We assign a distinct The image is scaled, since the initial label values would all appear same on the screen. The foreground objects initially assigned with binary values. If color to each label we obtain following results.

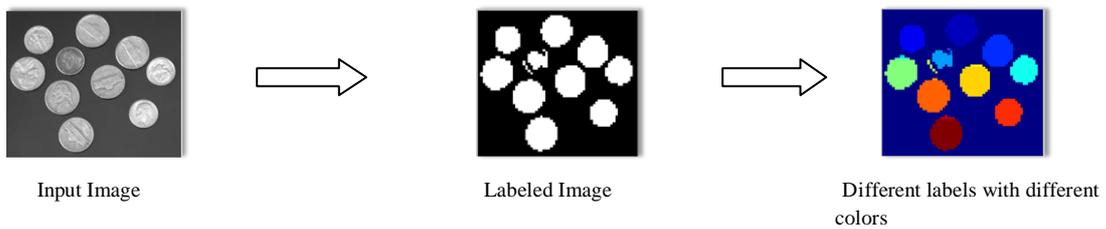


Fig. 15 Results after Labeling

2) *MATLAB Results:* The Fig.16 shows the MATLAB output of labeling method. There are three images displayed at the output. First image is simple binary image, second is primary labeled image and third is merged image. Following are the results of MATLAB simulation for different images.

From the output it can be observe that for primary labeled objects some portion at the upper side have different color and remaining portion have different color .This is because in primary labeling we are using multiple tracing and same object is divided in two parts while tracing. This causes same object to be labeled with two different labels.

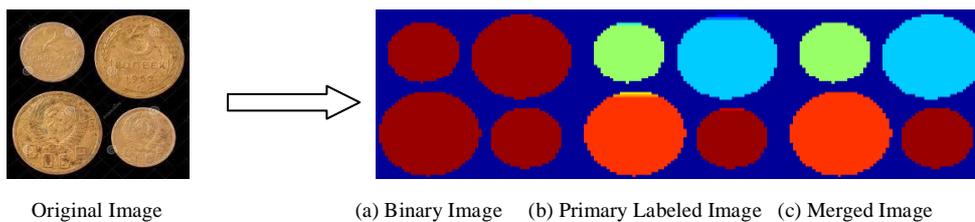


Fig. 16 Results of Two Pass Connected Component Labeling Algorithm of Image of Coins

D) Merging Model of Edge Map and Labeling

In this model merging of edge map and labeling is done for extracting features like area, centroid and bounding box of the objects present in the images. Here, two morphological operations that are erosion and dilation are used for the merging of edge mapping and labeling.

1) *System Generator Model for Merged Model of Edge Map and Labeling for Feature Extraction:*

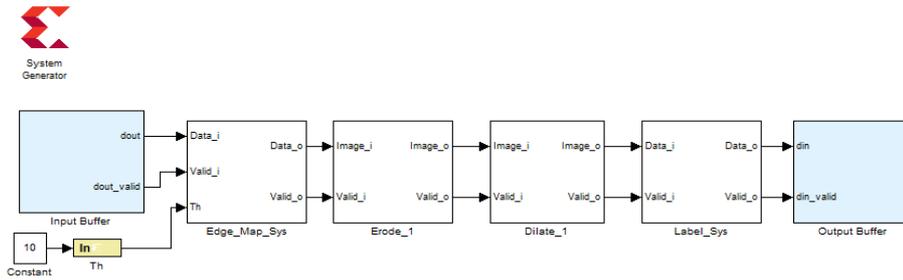


Fig. 17 Merged Top Model

Above Figure shows the System Generator Model for merging of Edge Map and Labeling algorithm. As shown in figure there are many blocks like Input buffer for Input to the Model, Edge Map block for the Edge Detection of images. Erode block is used to perform the morphological operation on edge detected images that means unwanted edges removed due to use of this erosion operator. Erosion operation not only removes the unwanted edges but also shrink another edges and this is the disadvantage of erosion. Therefore to overcome this disadvantage of erosion here dilate operator is used. Dilate block is used to gain the original shape of the image which changed due to use of erosion. Further Labeling block is used to perform the labeling and then merging operation of the images. And finally output buffer stores the output images and displayed on MATLAB. Morphological Operations and concept of Erosion and Dilation are described below in detail.

E) Results with FPGA Implementation

The figure below shows the system generator model result of Two Pass Labeling algorithm on hardware. Figure shows the system generator results for the image “Fruits” with Fig. (a) is the original input image. Fig. (b) is the simple labeled output image. Fig (c) shows output where different colors are assign to different objects. Fig.(d)is the final output after merging in second pass. The Hardware Model displays different features of the detected objects. The Blob Analysis block can be used to calculate Following Features for labeled regions in a binary image. The Blob Analysis block supports input and output variable size signals.

- 1) No of objects detected
- 2) Bounding Box
- 3) Centroid of detected object
- 4) Area of object

Fig.18 shows original input image which is taken as input in MATLAB. The image consists of multiple different fruits with different shapes.



Fig. 18 Original Image of fruits

Following are the FPGA output results where first image is binary image, second is primary labeled image applied with different colour for different labels, and third is final merged image.

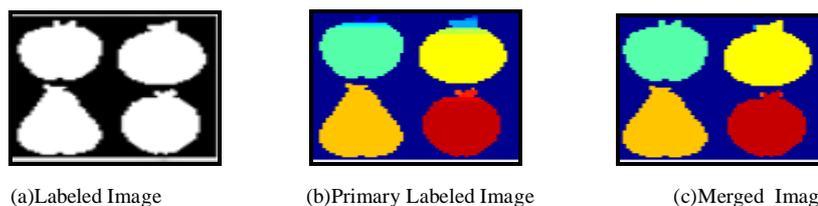


Fig.19 Output Images

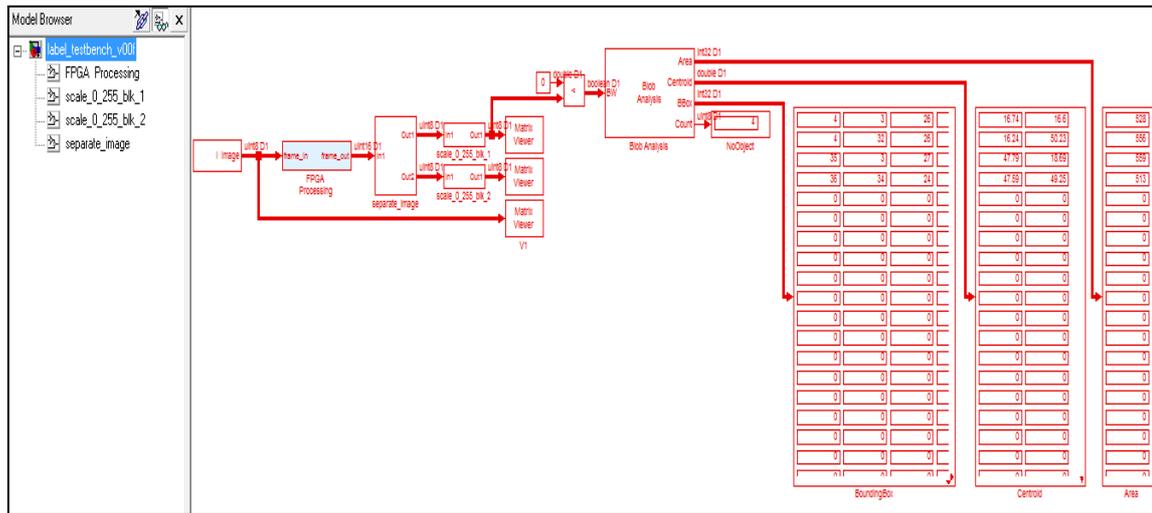


Fig. 20 Detected Objects Features

IV. Conclusions

The implementation of Image Edge Detection Algorithm, Labeling algorithm and merged model of Edge Map and Labeling algorithm on FPGA is present in this paper, and it is carried out successfully. The hardware and software implementation of system is found to be working properly. The MATLAB/Simulink models are used for the compatibility of implementing the image processing system on FPGA. The Virtex-5 ML506 evaluation board is user friendly which is used to implement this system. Implementation of Sobel Image edge detection algorithms using MATLAB/ SIMULINK is described. Edge Detection is done for still input images, inbuilt video and for Real time input from webcam of Laptop. By using two connected component labeling algorithm objects of images are labeled for distinguishing the objects. Therefore with the help of edge detection and labeling of images the various features i.e. area, bounding box and centroid are extracted.

V. Advantages

1. The primary advantages of the Edge detection using Sobel operator lie in its simplicity.
2. Due to simple computation of sobel edge detection it is less time consuming than canny edge detection.
3. Preserve important structural properties of image such as shape size, boundaries.
4. It required less hardware part.
5. Less memory consumption than canny edge detection.
6. Improved reliability and robustness of the system.
7. It requires very less power to get desired output.
8. Two pass connected component labeling algorithm can reduce the computational complexity.
9. It is Simple, easy to implement and suitable for low cost VLSI implementation.
10. It can break complex decision making process into a collection of simpler decisions.
11. Memory consumption is reduced for the labeling process of multiobject extraction.
12. It is user friendly system.
13. It can be used in handheld system.

VI. Limitations

For plane background better edge detection is possible. But in case of image with different backgrounds edges of desired object along with background of object are detected. Thus, the noises and unwanted edges still exist in the implementation results of this system. It is sensitive to the noise. The magnitude of the edges will degrade as the level of noise present in image increases. As a result, Sobel operator accuracy suffers as the magnitude of the edges decreases. Overall, the Sobel method cannot produce accurate edge detection with thin and smooth edge.

Acknowledgement

The author would like to express his sincere thanks to his Head of Department Dr. S.K. Shah for her valuable references and support throughout the project work. The author would also like to thank Prof. N.M. Wagdarikar for their valuable suggestions, support and co-operation.

The author would be grateful to his Principal, Dr. A.V. Deshpande and Vice Principal, Dr. K.R. Borole for their encouragement and guidance throughout the course.

References

- [1] O.R.Vincent,O.Folorunso, "A Descriptive Algorithm for Sobel Image Edge Detection" Proceedings of Informing Science & IT Education Conference (InSITE) 2009..
- [2] QingyiGu, Takeshi Takaki, and Idaku Ishii, "Fast FPGA-Based Multiobject Feature Extraction"IEEE Transactions on Circuits and Systems for Video Technology, vol. 23, no. 1, pp.30-45, January 2013
- [3] Kesheng Wu , EkowOtoo ,Kenji Suzuki, "Optimizing two-pass connected-component labeling algorithms" Pattern Anal Applic© Springer-Verlag London Limited 2008,pp. 117-135, March 2008M. Mueller, P. Karasev, I. Kolesov et al. "Optical Flow Estimation for Flame Detection in Video's". *IEEE Trans. On Image Proc.* Vol.22, No.7, July 2013.
- [4] H. Zhu and T. Shibata, Member, IEEE "A Real-Time Motion-Feature-Extraction VLSI Employing Digital-Pixel-Sensor-Based Parallel Architecture"IEEE Transactions on Circuits and Systems for Video Technology, pp. 1-13 , 2014
- [5] P. Vidya, S. Veni and K.A. Narayanankutty, "Performance Analysis of Edge Detection Methods on Hexagonal Sampling Grid", International Journal of Electronic Engineering Research, Volume 1 Number 4 (2009) pp. 313-328.